

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**

# **Previsão em tempo real de condições de tráfego em redes veiculares**

**Jorge Miguel Marques dos Reis**



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Rosaldo José Fernandes Rossetti

22 de Novembro de 2016



# **Previsão em tempo real de condições de tráfego em redes veiculares**

**Jorge Miguel Marques dos Reis**

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Daniel Augusto Gama de Castro Silva (PhD)

Arguente: Luís Paulo Gonçalves dos Reis (PhD)

Vogal: Rosaldo José Fernandes Rossetti (PhD)

---

22 de Novembro de 2016



# Resumo

Um dos maiores desafios propostos hoje em dia aos Sistemas de Transportes Inteligentes (ITS) é o apoio à redução do congestionamento das redes rodoviárias, devido ao elevado fluxo de veículos que se dirigem diariamente para as grandes áreas metropolitanas. O sistema de gestão de tráfego NEXT, desenvolvido pela empresa Armis ITS, permite atualmente realizar previsões, em tempo real, das condições de tráfego em redes rodoviárias. A sua implementação tem como base o recurso a técnicas de simulação (abordagem *model-driven*) e de *data mining* (abordagem *data-driven*), através de dados históricos recolhidos de sensores nas estradas.

As diferentes técnicas de previsão atualmente utilizadas geram, porém, resultados diferentes. Existe um módulo que, considerando o erro entre a previsão e a realidade, determina a eficiência de cada técnica através de uma percentagem de precisão (*accuracy*). A escolha da técnica apropriada dependerá, em grande parte, da capacidade do sistema em identificar a situação e contexto em que uma dada técnica se destaca das restantes. Este contexto está intimamente relacionado com o estado da rede, que pode ser caracterizado por diversas métricas que condicionam a circulação automóvel, tais como o dia e hora, as condições atmosféricas, a ocorrência de incidentes ou a topologia da rede. Assim, ter conhecimento desta relação entre estado da rede e técnica apropriada é particularmente imposto pela necessidade do sistema realizar previsões em tempo real. Desta forma, o principal problema a estudar é a identificação eficiente da melhor técnica de previsão a aplicar, de entre as técnicas implementadas, em casos e situações específicas.

A solução passa por estudar, analisar e implementar um módulo inteligente que, com base em técnicas de *machine learning*, permita decidir, considerando os métodos de previsão atualmente implementados no NEXT (Microsimulação e Rede Neuronal Artificial) e a sua precisão relativamente a cada estado da rede, qual a previsão mais adequada à situação.

Como resultados espera-se melhorar a eficiência do sistema de previsão, o que poderá ter um grande impacto na área de gestão e controlo de tráfego automóvel, tendo aplicação em sistemas de informação em tempo real. O desenvolvimento do projeto será orientado à sua integração no sistema NEXT que estará disponível em centros de controlo de tráfego da empresa Infraestruturas de Portugal (IP).



# Abstract

One of the biggest challenges proposed nowadays to Intelligent Transportation Systems (ITS) is the support to the reduction of traffic congestions due to the high flux of vehicles that travel daily to the major metropolitan areas. The traffic management system NEXT, developed by Armis ITS company, currently allows real-time forecasting of traffic conditions on road networks. Its implementation is based on the use of simulation techniques (model-driven approach) and data mining (data-driven approach), using historical data collected from road sensors.

However, the different forecasting techniques currently in use generate different results. There is a module that calculates the efficiency (percentage of accuracy) of each technique, considering the error between prediction and reality. Choosing the appropriate technique mainly depends on the system's ability to identify the situation and context where a given technique is distinguishable from the others. This context is closely related to the state of the network, which can be characterized by various metrics that influence the flux of vehicles, such as date and time, weather conditions, the occurrence of incidents or the network topology. So, be aware of the relationship between the state of the network and the suitable technique is particularly imposed by the need to make predictions in real time. This way, the main problem to be studied is the efficient identification of the best forecasting technique to apply in certain cases and situations.

The solution is to study, analyze and implement an intelligent module based on machine learning techniques, to choose the best forecasting algorithm, between the various algorithms currently implemented in NEXT, considering its accuracy for each state of the network.

As a result, it is expected to improve the efficiency of prediction algorithms, which can have a big impact on the management and control of road traffic, and can be applied in real time information systems. The project development will be guided to its integration into NEXT system, that will be available in traffic control centers of Infraestruturas de Portugal (IP).





# Agradecimentos

Em primeiro lugar, ao Prof. Rosaldo Rossetti, pela orientação e todo o apoio, suportados pela paixão que nutre pela área em estudo.

À Armis, por me receberem, e em especial ao José Macedo, Guilherme Soares e Ricardo Coelho, pela informação e dados disponibilizados, essenciais para a realização desta dissertação.

Aos meus amigos, colegas de curso e da TOPDOX, pelo incentivo.

À minha Mãe e à minha Namorada, por estarem ao meu lado nesta importante etapa da minha vida.

Ao meu Avô, que estará orgulhoso seguramente.

Ao meu Pai, o maior engenheiro que conheci na minha vida e que, decerto, fomentou o meu lado de engenheiro também. Esta dissertação é para ele.

Obrigado,  
Jorge Miguel Reis



*“Great things never came from comfort zones.”*

Rajat Dogra



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto . . . . .	1
1.2	Descrição do problema . . . . .	2
1.3	Motivação e Objetivos . . . . .	2
1.4	Contribuições esperadas . . . . .	3
1.5	Estrutura do relatório . . . . .	3
<b>2</b>	<b>Revisão Bibliográfica: Background</b>	<b>5</b>
2.1	Previsão de condições de tráfego . . . . .	5
2.1.1	Abordagem <i>model-driven</i> . . . . .	6
2.1.2	Abordagem <i>data-driven</i> . . . . .	6
2.2	Diagramas Fundamentais do fluxo de tráfego . . . . .	7
2.2.1	Diagrama fluxo-densidade . . . . .	7
2.2.2	Diagrama velocidade-densidade . . . . .	8
2.2.3	Diagrama velocidade-fluxo . . . . .	8
2.3	Congestionamento e eventos com influência no fluxo de tráfego . . . . .	9
2.4	Sensores de Tráfego . . . . .	10
2.4.1	Sensores <i>Inductive Loop</i> . . . . .	12
2.4.2	Formato <i>DATEX II</i> . . . . .	12
2.5	Aprendizagem Computacional . . . . .	12
2.5.1	Aprendizagem por Reforço . . . . .	12
2.5.2	<i>Q-Learning</i> . . . . .	13
2.5.3	<i>State-Action-Reward-State-Action</i> . . . . .	14
2.5.4	<i>Case-based reasoning</i> . . . . .	14
2.5.5	k-NN . . . . .	14
2.5.6	Árvore de Decisão . . . . .	15
2.6	Avaliação de métodos de classificação . . . . .	15
2.6.1	Matriz de Confusão . . . . .	15
2.6.2	<i>Cross-validation</i> . . . . .	16
2.7	Sumário . . . . .	16
<b>3</b>	<b>Revisão Bibliográfica: Trabalho Relacionado</b>	<b>17</b>
3.1	MAS-Ter Lab. . . . .	17
3.2	NEXT . . . . .	18
3.3	Algoritmos de Aprendizagem aplicados a ITS . . . . .	20
3.4	Seleção de Algoritmos utilizando <i>machine learning</i> . . . . .	20
3.5	Sumário . . . . .	21

## CONTEÚDO

<b>4</b>	<b>Abordagem Metodológica</b>	<b>23</b>
4.1	Formalização do Problema . . . . .	23
4.2	Descrição da fonte de dados . . . . .	24
4.2.1	Estrutura dos dados disponibilizados . . . . .	24
4.2.2	Dados históricos reais . . . . .	24
4.2.3	Caracterização dos dias . . . . .	25
4.2.4	Histórico de condições meteorológicas . . . . .	25
4.2.5	Histórico de incidentes . . . . .	25
4.2.6	<i>Thresholds</i> . . . . .	25
4.2.7	Histórico de previsões . . . . .	25
4.2.8	Previsões com <i>expert ANN</i> . . . . .	25
4.2.9	Análise descritiva dos dados . . . . .	26
4.3	Perspetivas de solução . . . . .	27
4.3.1	Arquitetura genérica do módulo <i>Expert</i> . . . . .	28
4.3.2	<i>Q-Learning</i> . . . . .	29
4.3.3	SARSA . . . . .	32
4.3.4	CBR . . . . .	33
4.3.5	Árvore de Decisão . . . . .	35
4.4	Cálculo do erro em previsões . . . . .	36
4.5	Metodologia de teste e análise de resultados . . . . .	36
4.6	Sumário . . . . .	37
<b>5</b>	<b>Implementação</b>	<b>39</b>
5.1	Ferramentas utilizadas . . . . .	39
5.2	Acesso a dados . . . . .	39
5.3	Módulo <i>expert</i> . . . . .	39
5.4	<i>Q-Learning</i> . . . . .	40
5.5	CBR . . . . .	41
5.6	Árvore de Decisão . . . . .	43
5.7	Sumário . . . . .	44
<b>6</b>	<b>Teste e Avaliação</b>	<b>45</b>
6.1	Dados utilizados para testes . . . . .	45
6.1.1	Notas importantes . . . . .	45
6.2	Teste 1 - Resultados de seleção do melhor método de previsão de tráfego . . . . .	46
6.2.1	Resultados utilizando Rede Neuronal Artificial . . . . .	46
6.2.2	Resultados do <i>expert Q-Learning</i> . . . . .	46
6.2.3	Resultados do <i>expert CBR</i> . . . . .	47
6.2.4	Resultados do <i>expert</i> Árvore de Decisão . . . . .	47
6.2.5	Discussão de Resultados . . . . .	48
6.3	Teste 2 - Resultados globais de previsão utilizando agentes <i>experts</i> . . . . .	49
6.3.1	Precisão máxima possível . . . . .	49
6.3.2	<i>Q-Learning</i> . . . . .	49
6.3.3	CBR . . . . .	50
6.3.4	Árvore de Decisão . . . . .	50
6.3.5	Discussão de Resultados . . . . .	51
6.4	Análise dos resultados obtidos . . . . .	52

## CONTEÚDO

<b>7</b>	<b>Conclusões e Trabalho Futuro</b>	<b>53</b>
7.1	Satisfação dos objetivos . . . . .	53
7.2	Trabalho futuro . . . . .	54
	<b>Referências</b>	<b>57</b>
<b>A</b>	<b>Matrizes de confusão de <i>cross-validation</i> para cada <i>edge</i> pelo método CBR</b>	<b>61</b>
<b>B</b>	<b>Matrizes de confusão de <i>cross-validation</i> para cada <i>edge</i> pelo método Árvore de Decisão</b>	<b>67</b>

## CONTEÚDO



# Lista de Figuras

2.1	As diferentes granularidades da simulação (da esquerda para a direita: macroscópica, microscópica e nanoscópica; mesoscópica no círculo) . . . . .	6
2.2	Diagrama fundamental do fluxo de tráfego. . . . .	7
2.3	Diagrama fluxo-densidade. . . . .	8
2.4	Diagrama velocidade-densidade. . . . .	9
2.5	Diagrama velocidade-fluxo. . . . .	9
2.6	Fontes de congestionamento (Estados Unidos da América) . . . . .	10
2.7	Sensores <i>inductive-loop</i> instalados numa via. . . . .	11
2.8	Esquema representativo de um sistema de <i>inductive-loop</i> . . . . .	11
2.9	Logótipo do formato Datex II. . . . .	12
2.10	Arquitetura da interação de um agente baseado em aprendizagem por reforço. . .	13
2.11	Ciclo fundamental do <i>Case-based reasoning</i> . . . . .	15
3.1	Representação de uma framework de análise de transportes. . . . .	18
3.2	Interface de dois componentes do sistema NEXT. . . . .	19
4.1	Gráfico com proporções de <i>traffic status</i> dos estados da rede no <i>dataset</i> . . . . .	26
4.2	Gráfico com proporção de melhor método de previsão para os estados da rede no <i>dataset</i> . . . . .	26
4.3	Gráfico de proporção de melhor método de previsão, agrupado por <i>traffic status</i> . .	27
4.4	Arquitetura do módulo <i>expert</i> . . . . .	28
4.5	Diagrama do algoritmo <i>Q-Learning</i> . . . . .	29
6.1	Gráfico de resultados de precisão para o teste 1. . . . .	48
6.2	Gráfico de resultados de precisão para o teste 2. . . . .	51

## LISTA DE FIGURAS

# Lista de Tabelas

4.1	Estrutura da matriz de confusão. . . . .	37
5.1	Estrutura da tabela Q. . . . .	40
5.2	Atributos caracterizadores de cada instância utilizada nos algoritmos k-NN e Árvore de Decisão. . . . .	44
6.1	Matriz de confusão dos resultados de seleção de método de previsão pela técnica Rede Neuronal Artificial. . . . .	46
6.2	Matriz de confusão dos resultados de seleção de método de previsão pela técnica <i>Q-Learning</i> . . . . .	47
6.3	Matriz de confusão dos resultados de seleção de método de previsão pela técnica CBR, utilizando <i>cross-validation</i> . . . . .	47
6.4	Matriz de confusão dos resultados de seleção de método de previsão pela técnica Árvore de Decisão, utilizando <i>cross-validation</i> . . . . .	48
6.5	Tabela com resultado das melhores previsões efetuadas pelos métodos disponíveis no sistema NEXT. . . . .	49
6.6	Matriz de confusão dos resultados de seleção de método de previsão pelo algoritmo <i>Q-Learning</i> , testado para dias 19 e 23 de Janeiro de 2014. . . . .	50
6.7	Tabela com resultado das previsões efetuadas pelo método de previsão selecionado pelo algoritmo <i>Q-Learning</i> , testado para dias 19 e 23 de Janeiro de 2014. . . . .	50
6.8	Matriz de confusão dos resultados de seleção de método de previsão pela técnica CBR, testado para dias 19 e 23 de Janeiro de 2014. . . . .	50
6.9	Tabela com resultado das previsões efetuadas pelo método de previsão selecionado pela técnica CBR, testado para dias 19 e 23 de Janeiro de 2014. . . . .	50
6.10	Matriz de confusão dos resultados de seleção de método de previsão pela técnica Árvore de Decisão, testado para dias 19 e 23 de Janeiro de 2014. . . . .	51
6.11	Tabela com resultado das previsões efetuadas pelo método de previsão selecionado pela técnica Árvore de Decisão, testado para dias 19 e 23 de Janeiro de 2014. . . . .	51

## LISTA DE TABELAS

# Abreviaturas e Símbolos

AD	Árvore de Decisão
API	<i>Application Programming Language</i>
ATM	<i>Active Traffic Management</i>
CBR	<i>Case-based reasoning</i>
FEUP	Faculdade de Engenharia da Universidade do Porto
GPS	<i>Global Positioning System</i>
IP	Infraestruturas de Portugal
ITS	<i>Intelligent Transportation System</i>
k-NN	<i>k-Nearest Neighbor</i>
LIACC	Laboratório de Inteligência Artificial e Ciência de Computadores
MAS	<i>Multi-Agent System</i>
MAS-Ter Lab.	<i>Laboratory for Multi-Agent base traffic and Transportation engineering research</i>
RL	<i>Reinforcement Learning</i>
SARSA	<i>State-Action-Reward-State-Action</i>
UP	Universidade do Porto
VCI	Via de Cintura Interna



# Capítulo 1

## Introdução

Este primeiro capítulo tem como intuito introduzir a presente dissertação, começando por abordar o seu enquadramento e descrição geral do problema. Posteriormente é revelada também a motivação para o desenvolvimento deste projeto e são também identificados os objetivos que se pretendem atingir. Por fim, é apresentada a organização geral deste documento.

### 1.1 Contexto

Atualmente, as condições de tráfego rodoviário representam um tema que afeta diariamente uma grande percentagem da população residente nas grandes áreas metropolitanas. Devido ao elevado fluxo diário de veículos para as cidades, é quase inevitável o congestionamento dos principais pontos de acesso, principalmente em determinadas horas do dia, correntemente denominadas "horas de ponta". No entanto, as condições de tráfego podem também ser influenciadas por outros fatores, tais como as condições meteorológicas, a topologia da rede, ou a ocorrência de acidentes ou eventos especiais numa determinada localização.

Neste contexto, a previsão em tempo-real das condições de tráfego torna-se muito interessante, sendo uma área que tem grande utilidade sobretudo no desenvolvimento de sistemas de navegação e informação aos viajantes, na gestão rodoviária efetuada nos centros de controlo de tráfego, e na atualização de painéis informativos presentes nas auto-estradas. Esta previsão é normalmente efetuada através do processamento de dados históricos, recolhidos de sensores nas auto-estradas e do uso de técnicas de simulação. A maior parte das estratégias para redução e prevenção de congestionamentos remetem para os Sistemas Inteligentes de Transportes (ITS) [RLT11].

Os sistemas de gestão ativa de tráfego (*Active Traffic Management Systems*) são considerados hoje em dia uma forma efetiva de monitorizar as redes rodoviárias continuamente e com custos reduzidos, permitindo a gestão da mesma [Sis12]. Para além de seguirem vários princípios dos Sistemas Inteligentes de Transportes, os ATMS pretendem maximizar a eficiência dos sistemas de transportes através da sua automatização, onde se poderá incluir a previsão a curto prazo das condições de tráfego.

### 1.2 Descrição do problema

Até ao momento, vários métodos de previsão de condições de tráfego foram já desenvolvidos. Utilizando como exemplo o sistema de previsão *NEXT*, desenvolvido pela empresa Armis ITS, o cálculo das previsões em tempo real é efetuado com recurso a técnicas de simulação (abordagem *model-driven*) e de processamento de dados (ou *data mining*) (abordagem *data-driven*). No entanto, as diferentes técnicas, aplicadas às mesmas situações, podem gerar resultados diferentes. O sistema é também capaz de, após receber os dados reais recolhidos pelos sensores rodoviários, calcular a precisão de cada técnica de previsão aplicada a uma data anterior com base no erro cometido.

Então, para o sistema de gestão de tráfego, o objetivo passará sempre pela escolha do melhor método de previsão. Porém, o principal problema prende-se com o facto da escolha da melhor técnica depender da situação e contexto específicos. Ou seja, pretende-se identificar, para cada contexto da rede, qual a técnica que se destaca das restantes, obtendo o menor erro na previsão. O contexto, ou situação, é caracterizado por atributos como a localização, o dia da semana e a hora, as condições atmosféricas, a ocorrência ou não de incidentes, entre outros. Estes atributos distinguem claramente um contexto de outro e assume-se que os algoritmos de previsão poderão ter um desempenho variável consoante o contexto em que se aplicam. Este problema, identificado pela Armis ITS, tentou ser solucionado inicialmente através do uso de uma Rede Neuronal Artificial (ANN), com resultados ainda aquém do desejável.

Este projeto foi proposto pelo Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC), pertencente à Faculdade de Engenharia da Universidade do Porto (FEUP), em colaboração da Armis ITS, empresa com conhecimento e experiência no desenvolvimento de Sistemas Inteligentes de Transportes.

### 1.3 Motivação e Objetivos

Analisando o tráfego das principais vias de acesso às cidades, neste caso em particular a Via de Cintura Interna (VCI), verifica-se a importância de compreender o tráfego nesta via de acesso desde e para a cidade do Porto. Através de técnicas de previsão baseadas no processamento de dados históricos recolhidos de sensores presentes nesta auto-estrada e de métodos de simulação, é possível prever as condições de tráfego a curto prazo. No entanto, como as diversas técnicas existentes produzem resultados diferentes consoante a situação e contexto em que se aplicam, surge a necessidade de incluir um agente *expert* capaz de, para cada estado da rede, identificar e executar em tempo real o método que apresenta os melhores resultados. Esta camada permite aumentar a eficácia da previsão, assim como reduzir custos computacionais desnecessários.

Desta forma, os principais objetivos a atingir nesta dissertação são:

- Estudo dos conceitos relacionados com previsão de tráfego rodoviário;
- Estudo e análise das metodologias e técnicas com potencial para contribuir na resolução do problema de decisão apresentado;



- Implementação de um protótipo do agente *expert*;
- Teste do componente desenvolvido;
- Análise e avaliação dos resultados;
- Escrita da presente dissertação.

### 1.4 Contribuições esperadas

O estudo e implementação de técnicas de *machine learning* aplicadas à escolha da melhor técnica de previsão de condições de tráfego apresenta-se como um ponto bastante importante na identificação final do estado do trânsito. Tendo consciência da necessidade de realizar previsões em tempo real, é de elevada importância que, assumindo como exemplo o sistema de gestão de tráfego desenvolvido pela Armis ITS e os métodos de previsão que o compõe, seja possível, através do contexto no momento, apresentar os resultados do método mais eficaz nesse caso. Dada a colaboração entre a Armis ITS e o LIACC-FEUP, a solução a ser desenvolvida deverá integrar o sistema de gestão de tráfego *NEXT*, que estará presente em centros de controlo de tráfego da Empresa Infraestruturas de Portugal (IP).

Espera-se que este projeto contribua assim para uma mais eficaz previsão das condições de tráfego, permitindo aos agentes responsáveis uma melhor e mais rápida atuação na manutenção do bom funcionamento das vias rodoviárias, melhorando as condições de circulação dos seus utilizadores.

### 1.5 Estrutura do relatório

Para além deste primeiro capítulo, onde é realizada uma introdução ao tema e contexto desta dissertação, este documento contém mais 6 capítulos principais. Neste primeiro é ainda formalizado o problema que deu origem à dissertação e são identificados os objetivos, metodologias e contribuições esperadas.

No capítulo 2, é realizada uma revisão bibliográfica dos temas que servem de base à realização desta dissertação, desde os conceitos base do tráfego rodoviário, algoritmos de aprendizagem computacional e técnicas de avaliação de métodos de classificação. No seguimento da revisão bibliográfica, são apresentados e discutidos, no capítulo 3, trabalhos relacionados e aplicações práticas dos conceitos anteriormente identificados em problemas semelhantes.

No capítulo 4 é feito um desenvolvimento do problema estudado nesta dissertação, procedendo-se à especificação da abordagem metodológica a ter em conta no seu desenvolvimento.

O capítulo 5 apresenta detalhadamente todo o processo de desenvolvimento dos agentes inteligentes. No capítulo 6 são apresentados os resultados provenientes dos testes aplicados, permitindo discutir e avaliar a solução implementada.

Por último, o capítulo 7 apresenta as principais conclusões obtidas após o desenvolvimento desta dissertação, assim como identificado o trabalho futuro que pode ser realizado a partir desta.

## Introdução

## Capítulo 2

# Revisão Bibliográfica: *Background*

Neste capítulo são apresentadas as principais definições e conceitos que servem de base à realização desta dissertação. Inicialmente é feita uma breve introdução ao estado da arte relativo à previsão de tráfego rodoviário e às abordagens utilizadas. Sobre o fluxo de tráfego são apresentados os seus diagramas fundamentais e a relação entre as três variáveis que caracterizam esse mesmo fluxo, e ainda os eventos que habitualmente o condicionam. Posteriormente são apresentados os conceitos associados à percepção do estado do tráfego, realizado através de sensores rodoviários e do formato *standard* europeu de comunicação de dados de tráfego: *Datex II*. Por fim é feita uma introdução às abordagens e algoritmos de aprendizagem computacional que podem ser aplicados na resolução do problema associado a esta dissertação, assim como técnicas de avaliação de resultados.

### 2.1 Previsão de condições de tráfego

A previsão de condições de trânsito tem um papel importante na gestão rodoviária, sobretudo pelo apoio no controlo e redução das filas e congestionamentos [Pin14]. Hoje em dia, com o objetivo de controlar e gerir as condições de tráfego rodoviário, várias estradas possuem dispositivos como sensores de *inductive loop*, controladores, sistemas de vídeo-vigilância e até dispositivos GPS em transportes públicos. Com recurso a estes sensores, instalados por exemplo na VCI, é possível obter uma elevada quantidade de dados caracterizadores das condições de tráfego em intervalos de poucos minutos. Através destes dados é possível realizar previsões de tráfego a curto prazo, com recurso a dois tipos de abordagem: a abordagem *model-driven*, constituída por técnicas de simulação, e a abordagem *data-driven*, que faz uso de técnicas de análise e processamento de dados [Bar15a].

Os dados históricos de sensores armazenados são utilizados nos modelos de análise de dados, mas são também fundamentais para a calibração dos parâmetros dos modelos computacionais usados em técnicas de simulação.

### 2.1.1 Abordagem *model-driven*

A abordagem *model-driven* procura construir modelos computacionais das redes rodoviárias que permitam representar o tráfego registado nas mesmas e a sua evolução, geralmente com recurso a técnicas de simulação. A rede modelada inclui cruzamentos, vias de rodagem, limites de velocidade, rotundas e sinais [BAR15b]. Desta forma, o sistema simula a movimentação de todas as entidades envolvidas segundo as características da via, incluindo as interações entre os veículos e peões. Assim, e com a capacidade de inserir informação em tempo-real no modelo, é possível obter previsões futuras das condições de tráfego.

Resumidamente, os modelos de simulação classificam-se em quatro categorias, segundo o nível de detalhe utilizado: macroscópicos, microscópicos, mesoscópicos e nanoscópicos [HLS07]. A modelação macroscópica apresenta o nível mais baixo de detalhe, focando-se em descrever as condições de tráfego com base na densidade por troço rodoviário. A abordagem microscópica fornece um nível de detalhe superior, simulando veículos individuais na rede, assim como as interações os mesmos. A abordagem nanoscópica apresenta um nível de detalhe ainda mais elevado, considerando a estrutura do veículo e suas características, tal como o tipo de motor, pneus, travões e consumo de combustível, por exemplo [Pin14]. O modelo mesoscópico surge da combinação entre os modelos macroscópicos e microscópicos, sendo o fluxo de tráfego apresentado com elevado detalhe, enquanto que as interações e comportamentos são apresentadas com menos detalhe. A figura 2.1 ilustra as diferentes granularidades que distinguem estes modelos.

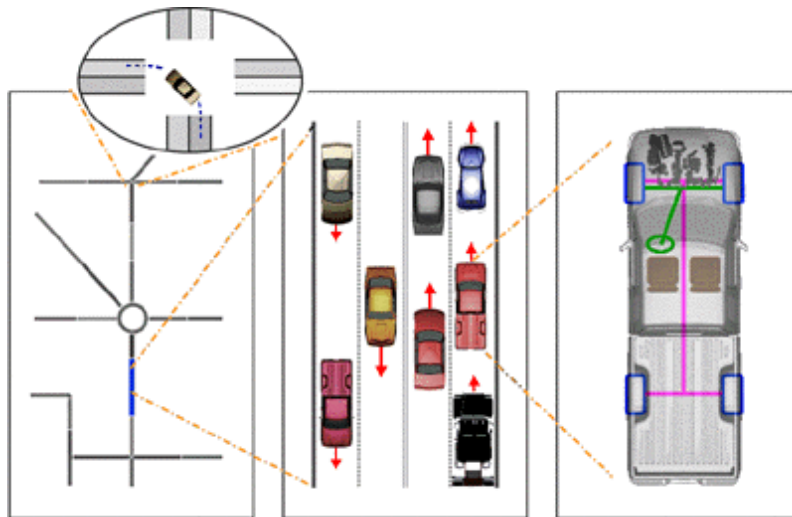


Figura 2.1: As diferentes granularidades da simulação (da esquerda para a direita: macroscópica, microscópica e nanoscópica; mesoscópica no círculo)

Fonte:[MR06]

### 2.1.2 Abordagem *data-driven*

No entanto, as soluções baseadas em modelos e simulação exigem processos de cálculo e calibração sendo por vezes pouco eficientes temporalmente. Contudo têm surgido soluções mais

rápidas, focadas na análise de dados históricos, seguindo uma abordagem *data-driven*. Estas permitem ainda incluir dados provenientes de outras fontes, tal como informação meteorológica ou registo de incidentes. As soluções mais frequentes utilizadas nestes modelos preditivos incluem métodos como k-NN, regressão, lógica fuzzy, redes bayesianas ou redes neuronais artificiais.

## 2.2 Diagramas Fundamentais do fluxo de tráfego

Um dos conceitos mais importantes no estudo do comportamento do tráfego rodoviário é a teoria do fluxo de tráfego, que consiste na aplicação de leis matemáticas, física e teoria da probabilidade ao tráfego automóvel. Os diagramas fundamentais do fluxo de tráfego são um conjunto de curvas que apresentam a relação entre as três variáveis associadas: o fluxo (veículos/hora), a velocidade (km/hora) e a densidade de tráfego (veículos/km). Estes diagramas são normalmente utilizados para prever a capacidade de uma via rodoviária tendo em conta a variação de regras e limites de velocidade, permitindo a exploração das relações apresentadas [MR06]. Todos estes gráficos, observáveis na figura 2.2 baseiam-se na equação essencial do fluxo de tráfego:

$$\text{fluxo} = \text{velocidade} \times \text{densidade}.$$

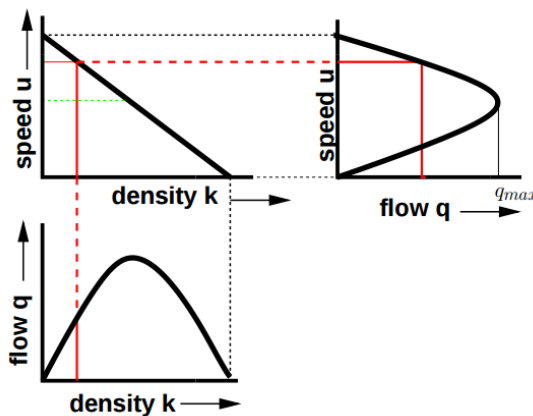


Figura 2.2: Diagrama fundamental do fluxo de tráfego.

Fonte:[MR06]

### 2.2.1 Diagrama fluxo-densidade

O diagrama fluxo-densidade permite determinar o fluxo de tráfego numa determinada via, apresentando um conjunto próprio de características:

- Quando a densidade tem o valor zero, o fluxo é também zero;
- A densidade e o fluxo aumentam com o aumento do número de veículos;

- Com o aumento contínuo do número de veículos, a dado momento atinge-se a densidade máxima, onde o fluxo é zero, dado que os veículos não se podem movimentar;
- Entre a densidade mínima e máxima existe um valor para o qual o fluxo é máximo, sendo esta relação representada por uma curva parabólica.

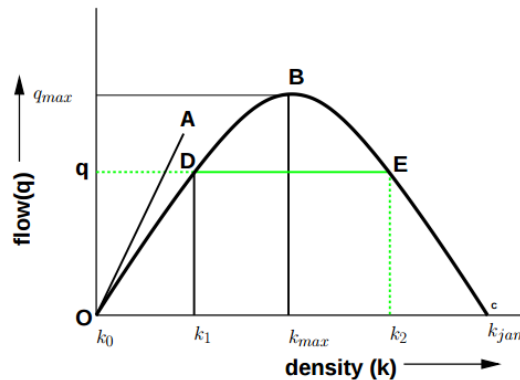


Figura 2.3: Diagrama fluxo-densidade.

Fonte:[MR06]

Na figura 2.3, onde se encontra representado o diagrama fluxo-densidade, é possível verificar as características descritas anteriormente. Assim, o ponto O representa a situação de densidade zero e o ponto C a de densidade máxima. O ponto B identifica a situação de fluxo máximo. O declive da tangente no ponto O (reta [OA]) corresponde à velocidade média em fluxo livre. Os pontos D e E demonstram a existência do mesmo fluxo em valores de densidade diferentes. Já a velocidade será superior em D relativamente a E, dado o menor número de veículos.

## 2.2.2 Diagrama velocidade-densidade

A partir dos valores de velocidade obtidos através da tangente à curva anterior, podemos representar a relação velocidade-densidade, observável na figura 2.4. Quando a densidade é igual a zero estamos perante um fluxo livre, enquanto que quando a velocidade é igual a zero significa que os veículos se encontram parados.

## 2.2.3 Diagrama velocidade-fluxo

Quando à relação entre velocidade e fluxo, representada na figura 2.5 observamos que o fluxo é zero quando não existem veículos na via ou quando existem mas trânsito se encontra parado. Verifica-se também que existem sempre duas velocidades diferentes para cada valor de fluxo, convergindo no ponto de fluxo máximo.

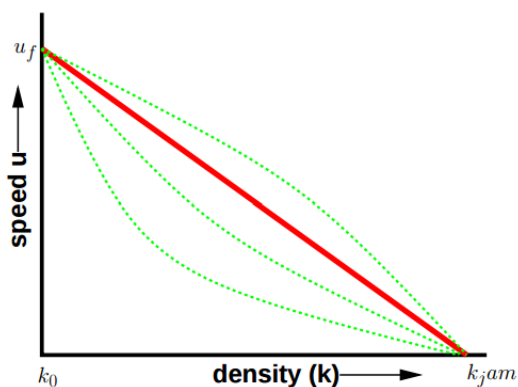


Figura 2.4: Diagrama velocidade-densidade.

Fonte:[MR06]

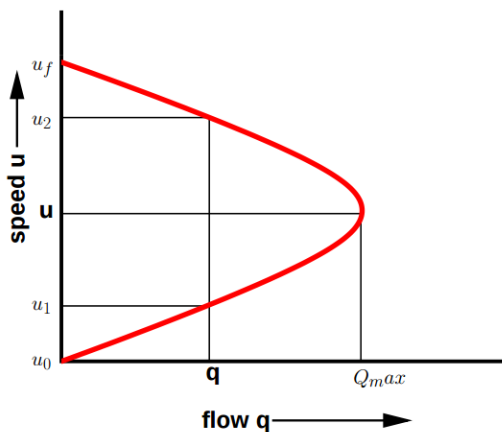


Figura 2.5: Diagrama velocidade-fluxo.

Fonte:[MR06]

## 2.3 Congestionamento e eventos com influência no fluxo de tráfego

Os congestionamentos rodoviários são eventos comuns nos grandes centros metropolitanos, e traduzem-se no excesso de veículos numa porção de via rodoviária e num dado momento, originando baixas velocidades de circulação. Muitas vezes tão baixa, que origina um andamento “pára-arranca”. Estudos indicam que os congestionamentos derivam de sete causas primárias, e que se dividem em três categorias [Uni15, Tra09]: eventos com influência direta no tráfego, variações do volume de veículos, e as características físicas das vias rodoviárias. Na primeira categoria, apresentam-se três causas:

1. Incidentes rodoviários, onde se incluem os acidentes entre veículos, as avarias, e o aparecimento de objetos que condicionem a circulação;

2. Zonas de obras, originando alterações físicas nas vias rodoviárias, incluindo redução do número de vias de circulação, alteração das vias ou mesmo encerramento temporário;
3. Condições Atmosféricas, que alteram o comportamento dos condutores, sobretudo quando são mais adversas, devido à redução da visibilidade e diminuição de aderência, obrigando à diminuição da velocidade e aumento das distâncias de segurança entre veículos.

Relativamente à categoria de variação do volume de veículos, são apresentadas duas causas primárias:

1. Flutuações no tráfego normal, havendo dias com maior volume de tráfego do que outros;
2. Eventos especiais, que originam um fluxo de tráfego superior ao normal nas imediações do local.

Quanto à última categoria, que se restringe às características físicas das vias de circulação, são apresentadas duas causas:

1. Dispositivos de controlo de tráfego, tais como passagens de nível e semáforos com tempos inapropriados;
2. *Bottlenecks* rodoviários, em que a capacidade física de determinados troços de uma estrada é diminuta, podendo ser incapaz de lidar com maiores volumes de tráfego.

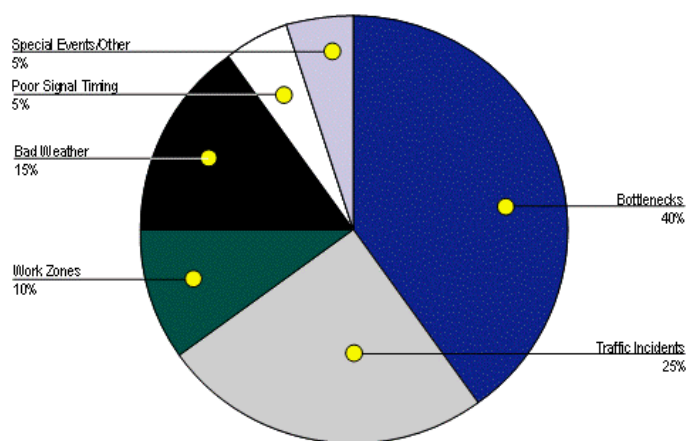


Figura 2.6: Fontes de congestionamento (Estados Unidos da América)

Fonte: *U.S. Department of Transportation* [Uni15]

## 2.4 Sensores de Tráfego

A principal fonte de dados dos sistemas de informação de tráfego são os sensores rodoviários [BSM00, KMG06]. Estes sensores são aparelhos com capacidade de recolher continuamente



dados representativos do estado do tráfego rodoviário na área em que se encontra. Existem vários tipos de sensores, de entre os quais se destacam as câmaras de vídeo-vigilância, os *floating cars* e os sensores de *inductive-loop*, sendo estes últimos explicados de forma mais detalhada na subsecção seguinte.

As câmaras de vídeo-vigilância, instaladas nas estradas, são utilizadas para capturar imagens em tempo real, que permitem analisar o estado do trânsito na zona onde se encontra [LRB09].

*Floating Cars* são veículos que incluem um recetor GPS e um transmissor GSM/GPRS, transmitindo dados acerca da circulação do veículo [ZCWY14]. Desses dados destacam-se a velocidade e localização do veículo a uma certa data e hora de captura.

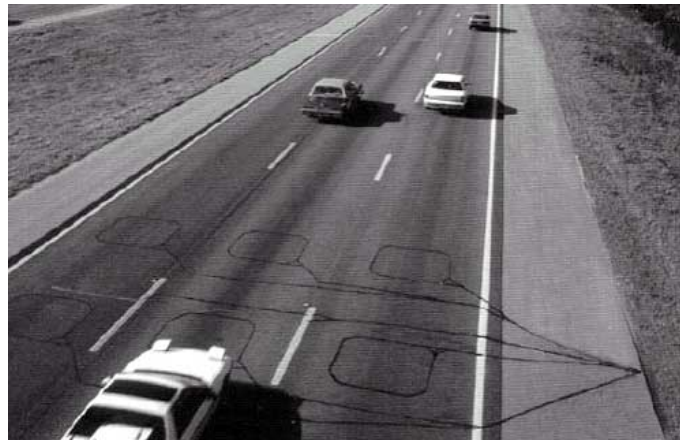


Figura 2.7: Sensores *inductive-loop* instalados numa via.  
Fonte: *U.S. Department of Transportation*

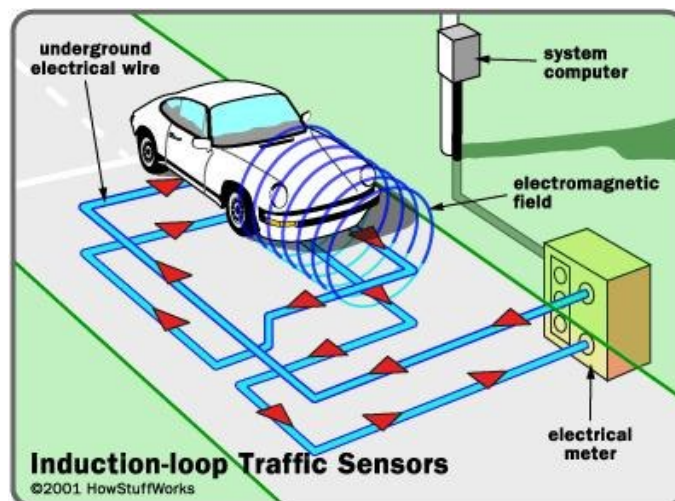


Figura 2.8: Esquema representativo de um sistema de *inductive-loop*.  
Fonte: HowStuffWorks

### 2.4.1 Sensores *Inductive Loop*

Os sensores *inductive loop* [Law01, KMG06] são espiras magnéticas instaladas no pavimento de estradas, tal como os representados na figura 2.7. Têm como função detetar a passagem de veículos, funcionando por capturas durante curtos períodos de tempo, gerando dados que indicam o número de veículos, a velocidade média, o espaçamento entre eles e a taxa de ocupação do sensor durante o tempo de captura. É associada ainda a localização do sensor, a data e hora da captura. O funcionamento do sistema está representado na figura 2.8.

### 2.4.2 Formato *DATEx II*

O formato Datex II [ECT13] é um protocolo *standard* europeu para troca de dados de tráfego, desenvolvido no âmbito do Plano de Ação ITS, promovido pela Comissão Europeia. O Datex (*data exchange*) permite a troca de dados entre os centros de controlo e gestão de tráfego, e entre os centros de informação de tráfego e os prestadores de serviços.



Figura 2.9: Logótipo do formato Datex II.

## 2.5 Aprendizagem Computacional

Do ponto de vista da solução que se pretende implementar, foi realizada uma revisão da literatura relativa a métodos de aprendizagem. São, de seguida, apresentados os algoritmos mais referenciados e com melhores perspetivas de sucesso quando integrados na solução pretendida neste trabalho.

### 2.5.1 Aprendizagem por Reforço

Aprendizagem por reforço (em inglês, *Reinforcement Learning*) é o problema enfrentado por um agente que aprende o seu comportamento através de tentativa e erro, quando em interação com um ambiente dinâmico [KLM96, McC15]. Este é um método de aprendizagem onde se procura maximizar uma medida de desempenho baseada nos reforços recebidos através da interação do agente com o ambiente envolvente. Normalmente este tipo de aprendizagem é utilizado quando não se dispõe de um modelo concreto, sendo impraticável o uso de aprendizagem supervisionada.

Assim, o seu objetivo passa exploração de diferentes ações para pesquisa do espaço de possíveis políticas, permitindo encontrar uma política de atuação ótima e estimar a melhor ação para cada estado.

O diagrama 2.10 representa o ciclo de interação do agente baseado num algoritmo de aprendizagem por reforço.

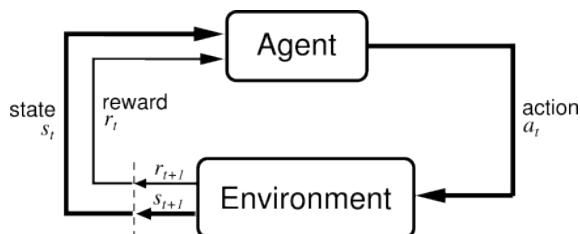


Figura 2.10: Arquitetura da interação de um agente baseado em aprendizagem por reforço.

Fonte:[SB98]

É também necessário um balanceamento entre os processos denominados *exploration* e *exploitation*. O primeiro, permite ao agente experimentar e explorar o espaço de possíveis ações. Caso o agente não o faça, corre o risco de permanecer numa política subótima. O segundo, por outro lado, consiste no refinamento do conhecimento através da aplicação apenas da ação que maximiza o reforço, ou seja, considerada ótima.

### 2.5.2 *Q-Learning*

Um dos maiores avanços realizados na aprendizagem por reforço foi o desenvolvimento do algoritmo *Q-Learning* por Christopher Watkins [Wat89, WD92], um método de diferença temporal *off-policy*, tal que o valor de utilidade  $Q$  de um par estado-ação é calculado através da soma dos reforços obtidos ao executar uma dada ação sobre um determinado estado. Normalmente estes valores de utilidade para cada par são armazenados numa tabela para contínua atualização. Neste algoritmo o agente tem como objetivo aprender uma política ótima através da sua interação com o meio, traduzindo-se em sequências estado-ação com o respetivo reforço. Assim, encontrando-se o agente num estado  $s$  e realizando uma ação  $a$ , recebe um a recompensa  $r$  e transita para o estado  $s'$ .

Na sua forma mais simples, de nome *one-step Q-learning* [SB98], é definido pela expressão:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (2.1)$$

onde onde  $s_t$  representa o estado num momento  $t$ ,  $a_t$  representa a ação tomada no instante  $t$ ,  $R_{t+1}$  é o custo por essa decisão,  $\alpha$  é a taxa de aprendizagem e  $\gamma$  o fator de desconto para transições futuras. Desta forma, a função ação-valor  $Q$  alvo de aprendizagem aproxima-se da função ação-valor ótima, independentemente da política seguida.

### 2.5.3 *State-Action-Reward-State-Action*

*State-Action-Reward-State-Action* (SARSA), é um algoritmo de aprendizagem introduzido por Rummery e Nirankan em 1994 [RN94], inicialmente denominado *On-line Q-Learning*. É um algoritmo *on-policy*, e o nome deve-se ao facto do valor  $Q$  depender do estado atual, da ação tomada e do reforço obtido, assim como do próximo estado alcançado e da próxima ação que o agente tomará neste novo estado.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (2.2)$$

Como definido em 2.2, o valor da função  $Q$  para um determinado estado-ação é atualizado pelo reforço obtido, influenciado por uma taxa  $\alpha$  de aprendizagem. Desta forma, o valor  $Q$  representa o reforço obtido tomando uma ação  $a$  no estado  $s$ , somando ainda um futuro reforço obtido a partir do próximo par estado-ação.

### 2.5.4 *Case-based reasoning*

*Case-based reasoning* (CBR) é uma técnica de resolução de novos problemas através da adaptação de soluções utilizadas para resolver problemas anteriores semelhantes [AP94]. Este processo foi desenvolvido inicialmente por Roger Schank no início dos anos 80 [Sch82], tendo sido desenvolvido o primeiro sistema baseado neste modelo em 1983 por Janet Kolodner [Kol92].

Dada a inspiração no modelo humano, a ideologia presente no *case-based reasoning* faz com que este seja também considerado um método de resolução de problemas gerais do dia-a-dia, para além de um poderoso método de decisão computacional.

Por vezes assumido como um algoritmo ou tecnologia, o CBR é, segundo Watson [Wat99], uma metodologia de aprendizagem. Metodologia essa que é habitualmente descrita como um ciclo de quatro etapas que precedem a construção do *case base*:

1. Recuperar casos semelhantes ao novo problema do *case base*;
2. Reutilizar a solução proposta pelos casos semelhantes;
3. Rever ou adaptar a solução ao novo problema, se necessário;
4. Reter o novo caso e respetiva solução após ser validada.

Este ciclo e respetivas interações estão retratados na figura 2.11.

### 2.5.5 *k-NN*

O algoritmo *k-Nearest Neighbors* é um método de aprendizagem supervisionada, utilizado em problemas de classificação e regressão [Alt92]. Em ações de classificação, como irá ser utilizado no desenvolvimento desta dissertação, um novo caso é classificado tendo em conta a classe da maioria dos  $k$  elementos de treino mais próximos num espaço  $n$ -dimensional, sendo  $n$  o número de atributos que caracterizam um caso. O parâmetro  $k$  representa o número de vizinhos mais

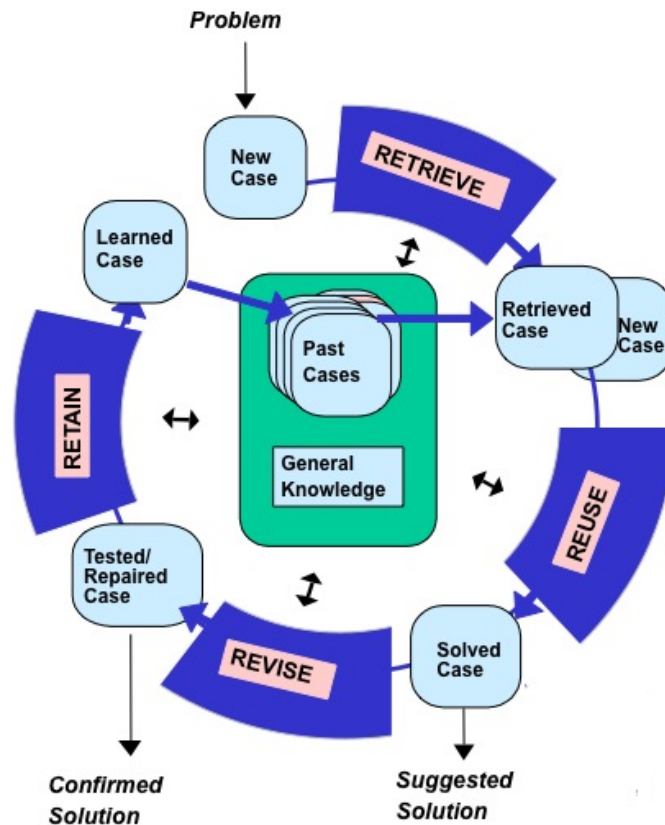


Figura 2.11: Ciclo fundamental do *Case-based reasoning*  
 Fonte:[And12]

próximos a recuperar, sendo que se for 1, será atribuído ao novo caso a mesma classe do vizinho mais próximo.

### 2.5.6 Árvore de Decisão

A Árvore de decisão [Qui86, SL90] é um modelo de inferência indutiva, representando uma tabela de decisão em forma de árvore, e é construída por algoritmos como o ID3 e o C4.5. Estas árvores são criadas a partir de um conjunto de dados de treino com elementos classificados, utilizando o conceito de entropia. Tal como a tabela de decisão, é uma forma de apresentação de regras sobre as quais são obtidas as classificações.

## 2.6 Avaliação de métodos de classificação

### 2.6.1 Matriz de Confusão

A matriz de confusão [KP98], utilizada essencialmente na área de *machine learning*, é uma tabela que permite visualizar a performance de um algoritmo de classificação [Ste97, Pow07]. Esta tabela apresenta informação sobre as classificações previstas e as reais, permitindo avaliar,

para além da precisão, mais 4 medidas que correspondem às percentagens de acerto na previsão e às percentagens de elementos de cada classe que foram bem classificados.

### 2.6.2 *Cross-validation*

*Cross-validation* [Koh95] é uma técnica utilizada para avaliar a precisão e erro de um algoritmo preditivo com base num conjunto de dados a ser utilizado para treino e teste. O conjunto é então dividido em  $k$  subconjuntos de igual tamanho, sendo que um deles é utilizado para teste do modelo preditivo, enquanto que os restantes serão utilizados para treinar o modelo. Este processo repete-se por  $k$  rondas, alternando o subconjunto de teste, permitindo utilizar todos os dados disponíveis para treino e teste do modelo.

## 2.7 Sumário

Neste capítulo foram revistos os principais conceitos que servem de base a este trabalho de dissertação. Desde as principais abordagens utilizadas na previsão, passando pelos conceitos fundamentais associados ao fluxo de tráfego e congestionamento. Foram ainda apresentados os sensores utilizados na recuperação de dados históricos das vias rodoviárias, em especial foco para os sensores *inductive loop*, utilizados como fonte de dados dos algoritmos de previsão desenvolvidos pela Armis ITS, e que seguem o formato Datex II, promulgado pela Comissão Europeia. Foram ainda descritos os algoritmos de aprendizagem relevantes para este projeto, nomeadamente *Q-Learning*, *State-Action-Reward-State-Action* (SARSA), *case-based reasoning* (CBR), k-NN e Árvores de Decisão. Por último, uma rápida referência à avaliação de algoritmos de classificação, nomeadamente a matriz de confusão e a técnica de teste e avaliação *cross-validation*.

## Capítulo 3

# Revisão Bibliográfica: Trabalho Relacionado

Após a análise e compreensão dos principais conceitos associados ao fluxo de tráfego e à sua previsão, assim como algoritmos de aprendizagem computacional que poderão ser importantes no âmbito desta dissertação, este capítulo tem como objetivo apresentar uma revisão do trabalho já realizado nesta área. Dado o problema específico, são apresentados trabalhos relacionados com duas áreas. Primeiro, são apresentados trabalhos na área de previsão de tráfego rodoviário que, por constituírem a base e parte do desenvolvimento do sistema de gestão de tráfego pertencente à Armis ITS, servem também de base ao desenvolvimento da presente dissertação. Posteriormente são apresentadas também aplicações dos algoritmos de aprendizagem anteriormente referenciados em áreas relacionadas com Sistemas Inteligentes de Transportes e na seleção de algoritmos apropriados.

### 3.1 MAS-Ter Lab.

A plataforma MAS-Ter Lab. (Laboratory for MAS-based Traffic and Transportation Engineering Research) [ROB07, RFBO08, FERO08] trata-se de um sistema multi-agente integrado que representa uma abordagem metodológica para avaliação de soluções inteligentes de transportes atuais através da metáfora de agentes. Nesse sentido, o domínio de aplicação é conceptualizado em termos de agentes, sendo identificados três subsistemas, que são eles próprios sistemas multi-agente: o *real world* (mundo real), um *virtual domain* (domínio virtual) e o *control strategies inductor* [FG09, KSPRG]. O subsistema *real world* representa o sistema real de transportes em áreas urbanas, onde se encontram e interagem os componentes físicos, tais como os veículos, os sistemas de controlo de tráfego e as soluções ITS. O *virtual domain* corresponde à replicação destes componentes, modelados sobre a forma de agentes, que pretendem emular o comportamento individual dos componentes físicos no *real world*. Por fim, o *control strategies inductor* é um subsistema constituído por agentes especialistas, tanto humanos como computacionais, que observam a população sintética do *virtual domain* e podem intervir e realizar experimentações diretamente



com esta, aplicando políticas de coordenação com o intuito de melhorar o desempenho global. É ainda possível ajustar parâmetros operacionais do mundo real para refletir as políticas testadas. A interação entre os três subsistemas referidos é dinâmica e iterativa, o que permite intervenções em tempo real no mundo real (*real world*) [AFR08].

Uma representação da integração destes conceitos pode ser observada na figura 3.1.

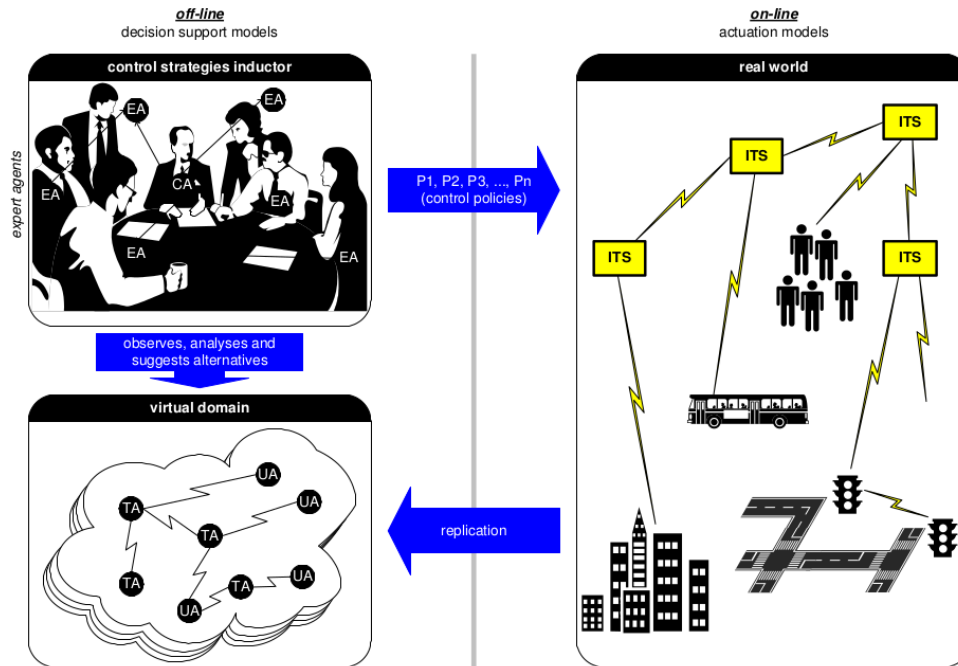


Figura 3.1: Representação de uma framework de análise de transportes.

Fonte:[ROB07]

## 3.2 NEXT

O NEXT, sistema produzido pela Armis ITS [Arm16], trata-se de uma solução para a análise, simulação e previsão de dados de tráfego, baseando-se no modelo da framework [MAS-Ter Lab](#). Este sistema inclui atualmente técnicas e algoritmos de previsão de tráfego baseados em abordagens *model-driven* e *data-driven*, nomeadamente simulação micro e macroscópica e redes neurais artificiais. Com recurso a simuladores de redes rodoviárias, possibilita a representação de uma rede real de tráfego através de redes virtuais. Este serviço apresenta várias funcionalidades de previsão e apoio ao planeamento, nomeadamente os componentes: *Monitor*, *Predictor*, *Advisor*, *Planner* e *Auditor*.

O sistema permite assim monitorizar em tempo real o estado do tráfego, prever o estado de tráfego futuro até duas horas em intervalos de cinco minutos, e auditar a precisão das previsões com base nos resultados reais recebidos posteriormente. Integra ainda funcionalidades de planeamento, com base no estado de tráfego em tempo real.



## Revisão Bibliográfica: Trabalho Relacionado

Em [ZRC14, Zai14] é proposto o desenvolvimento de uma *dashboard* que melhora a capacidade de visualização do estado da rede, tendo sido este projeto desenvolvido no âmbito do sistema NEXT.



(a) Funcionalidade *Planner*



(b) Funcionalidade *Auditor*

Figura 3.2: Interface de dois componentes do sistema NEXT.

### 3.3 Algoritmos de Aprendizagem aplicados a ITS

Existem vários exemplos da aplicação de algoritmos de aprendizagem a sistemas inteligentes de transportes. Após uma revisão bibliográfica do tema, embora não se tenha encontrado nenhum exemplo de aplicação de aprendizagem em um semelhante ao proposto nesta dissertação, existem aplicações em diferentes situações relacionadas com controlo de tráfego rodoviário.

Um exemplo recente é o de um sistema de gestão de tráfego multi-modal [SSAEA15] baseado no algoritmo *Q-learning*, tendo em conta o contexto para fornecer melhores decisões. A aprendizagem neste sistema tem como objetivo dotá-lo da capacidade de adaptar dinamicamente a escolha do melhor meio de transporte baseado em *feedbacks* fornecidos pelos passageiros. Outro exemplo descrito no artigo [SJ11] demonstra a capacidade de um sistema baseado em *Q-Learning* de se adaptar à variação das condições de tráfego para gestão dos tempos semafóricos em intersecções rodoviárias.

Relativamente ao algoritmo *State-Action-Reward-State-Action* (SARSA), a principal utilização encontrada em Sistemas Inteligentes de Transportes é em controlo de semáforos rodoviários. No recente artigo [JM15] os algoritmos SARSA e Q-Learning são estudados e testados em aprendizagem aplicada à gestão semafórica em intersecções de quatro vias.

Em termos de utilização do algoritmo *case-based reasoning* (CBR), foram também encontradas utilizações em controlo de semáforos rodoviários em intersecções urbanas, como por exemplo nesta tese de mestrado [And12]. No entanto, no artigo [BFAC13] é proposta uma abordagem para sugestão de itinerários baseados nas necessidades e preferências dos utilizadores, resultante da integração de CBR com *Choquet integral*. O artigo [BLZH15] apresenta um estudo sobre a aplicação de CBR em sistemas de planeamento de meios de transporte.

### 3.4 Seleção de Algoritmos utilizando *machine learning*

Embora esta dissertação se contextualize na área de Sistemas Inteligentes de Transportes, o principal caso de estudo é a aplicação de métodos de aprendizagem computacional à seleção de algoritmos dinamicamente. A seleção de algoritmos é um problema difícil e estudado há várias décadas [Ric76], e serão apresentados de seguida alguns projetos em que técnicas de *machine learning* foram utilizadas como soluções para este problema.

No artigo [LL00] é descrita a adaptação do algoritmo *Q-learning* para resolução do problema de seleção de algoritmos. O principal objetivo desta implementação é a redução do tempo total de execução na resolução de um determinado problema.

No artigo [ACMR06] é apresentado um protótipo de uma ferramenta que utiliza aprendizagem por reforço para seleção de algoritmos em tempo de execução, respondendo a alterações no estado do programa e ambiente computacional.

No artigo [KGM12], são apresentados vários métodos de *machine learning* utilizados para seleccionar o melhor algoritmo, de um conjunto disponível, em problemas de pesquisa. O primeiro

utilizado foi o *case-based reasoning* (CBR). Neste caso, foi utilizado o algoritmo k-NN para indexação e recuperação de casos, com  $k$  de valores 1, 3, 5 e 10. Este método foi implementado com recurso ao algoritmo IBk, que implementa o k-NN, presente no pacote de *software open-source* Weka. Como a seleção de algoritmos representa um caso de classificação, foram testados também vários métodos de classificação neste problema, tais como Árvores de decisão, tabela de decisão, redes bayesianas, entre outros. A terceira metodologia passa pela utilização de regressão. Esta permite selecionar o melhor algoritmo prevendo a performance que cada um terá em certas condições. A desvantagem é que em vez de executar a aprendizagem uma vez por problema, é necessário executá-la para cada algoritmo do conjunto para um único problema. Por último, são também apresentados métodos de aprendizagem estatística relacional, abordagem que é identificada como promissora por ser a que apresenta um modelo mais intuitivo para os humanos.

O artigo [Fin98] apresenta uma abordagem baseada na análise estatística de performances passadas na seleção de métodos de resolução para novos problemas.

### 3.5 Sumário

Neste capítulo são identificados, primariamente, os dois projetos intimamente relacionados com o projeto a desenvolver nesta dissertação: MAS-Ter Lab. e NEXT. É apresentada também uma revisão bibliográfica sobre a aplicação de algoritmos de aprendizagem a ITS, onde se pôde verificar que a maior parte da sua utilização parece resumir-se a planeamento de itinerários e meios de transporte e, em maior quantidade, sistemas de controlo de semáforos rodoviários. Em termos de aprendizagem aplicada à seleção de algoritmos foram identificados alguns casos de estudo bastante interessantes e que se assemelham um pouco ao contexto desta dissertação, contudo o objetivo final é distinto.

## Revisão Bibliográfica: Trabalho Relacionado

## Capítulo 4

# Abordagem Metodológica

Tendo em conta a introdução aos conceitos e revisão de trabalho relacionado nos capítulos anteriores, pretende-se agora, neste capítulo, apresentar a abordagem metodológica idealizada para o desenrolar desta dissertação e cumprimento dos objetivos propostos. Inicialmente, é realizada a formalização do problema que deu origem a esta dissertação. De seguida é apresentada a estrutura dos dados disponibilizados pela Armis ITS, segundo as tabelas e atributos respetivos. Posteriormente são então apresentadas as perspetivas de solução e identificados os passos a realizar no desenvolvimento dos agentes *experts*. Por fim é apresentada a metodologia de análise e avaliação dos resultados obtidos.

### 4.1 Formalização do Problema

Como introduzido no primeiro capítulo deste documento, a previsão em tempo real das condições de tráfego rodoviário assume um papel muito importante nos atuais Sistemas Inteligentes de Transportes (ITS), prevendo o estado da rede rodoviária para períodos próximos e assim apoiando significativamente a gestão dessas mesmas redes. Atualmente a previsão depende essencialmente de duas abordagens: *model-driven* e *data-driven*. A empresa Armis ITS, com a colaboração da qual esta dissertação foi desenvolvida, possui hoje em dia dois métodos completamente implementados e em funcionamento no sistema NEXT: Microsimulação (que segue a abordagem *model-driven*), e uma Rede Neuronal Artificial (abordagem *data-driven*).

No entanto, após análise, concluiu-se que os resultados de previsão diferiam de método para método, e pretende-se que o sistema apresente o resultado mais próximo da realidade. Assim, assumiu-se que cada método tenderá a ser mais eficiente em determinados estados da rede, sendo que estes estados se caracterizam pela localização ou tipo de via, hora, dia da semana, condições atmosféricas, ocorrência de incidentes, entre outras características. Um exemplo de fácil percepção é a ocorrência de um acidente numa via. Sendo que o método por Rede Neuronal Artificial se baseia em dados históricos (e neste momento apenas em cada ponto de medição individual), percebe-se que o método de Microsimulação deverá ser mais eficiente neste caso pois simula concretamente a propagação das consequências com base num modelo para o estado da rede atual. E

foi este o primeiro caso a ser pensado e testado na Armis na seleção do método de previsão ideal: caso exista um incidente é utilizada Microsimulação, caso contrário é usada a Rede Neuronal Artificial. No entanto os resultados não foram satisfatórios.

Assim, o próximo passo, e inspiração para esta dissertação, passa por delinear o desenvolvimento de um módulo inteligente baseado em *machine learning* com a capacidade de, tendo em conta o estado da rede rodoviária e contexto no momento, determinar em tempo real o método de previsão mais preciso e adequado à situação. Para realizar a aprendizagem desde módulo inteligente serão utilizados os resultados de precisão para cada algoritmo num determinado espaço de previsões efetuados pela Armis, com recurso à sua plataforma NEXT.

## 4.2 Descrição da fonte de dados

Para a evolução desta dissertação é de extrema importância a utilização de dados reais relativos ao estado da rede. Graças à parceria existente entre o LIACC-FEUP e a empresa Armis ITS, e à relação existente entre esta última e a Infraestruturas de Portugal (IP), é possível obter um vasto conjunto de dados relativos às medições de tráfego (volume de veículos, velocidade média e ocupação média do sensor) dos sensores *inductive loop* instalados na VCI, agregados em intervalos de 5 minutos. Estes dados são processados pela Armis e armazenados numa base de dados Microsoft SQL Server, num formato próprio.

Para a realização desta dissertação foram utilizados todos os dados disponibilizados, que contemplam dados históricos e previsões completas entre os dias 1 e 24 de Janeiro de 2014, para 27 *edges*, para além de outros dados complementares.

O conceito de *edge*, que será amplamente utilizado nesta dissertação, significa um troço de via rodoviária onde normalmente se encontra um ponto de medição sensorial e para a qual são realizadas previsões de condições de tráfego.

### 4.2.1 Estrutura dos dados disponibilizados

Os dados disponibilizados pela Armis e relevantes para esta dissertação, encontram-se estruturados em 8 tabelas principais: *HistData*, *DateType*, *HistWeatherData*, *Incidents*, *Thresholds*, *HistForecastedANNDData*, *HistForecastedMicroData* e *HistForecastedData*. Nas próximas subsecções serão identificados os dados utilizados e disponibilizados nestas tabelas.

### 4.2.2 Dados históricos reais

A tabela *HistData* representa os dados históricos reais resultantes de medições de sensores *inductive loop* na VCI, de onde se destacam a data de gravação, o volume de veículos, a velocidade média, a ocupação e o *traffic status* resultante. Caracterizados em intervalos de 5 minutos, e numa determinada *edge* (ponto do mapa com sensores), estes dados permitem saber os valores reais do tráfego automóvel num dado momento, permitindo posteriormente determinar a precisão dos algoritmos de previsão.

#### 4.2.3 Caracterização dos dias

A tabela *DateType* é aquela onde se encontram os dados de caracterização de cada dia. Nela estão localizadas informações sobre uma determinada data, tais como o dia da semana, o dia do ano, a semana do ano, se é feriado, se é véspera de feriado, se é um dia de ponte, ou até se faz parte de um período de férias escolares.

#### 4.2.4 Histórico de condições meteorológicas

A tabela *HistWeatherData* armazena o histórico de condições atmosféricas, disposto em intervalos de 30 minutos e por cidade (atualmente apenas da cidade do Porto). Estes dados são também de vital importância na previsão do estado do tráfego rodoviário.

#### 4.2.5 Histórico de incidentes

O histórico de incidentes, também utilizado nos algoritmos de aprendizagem, está contido na tabela *Incidents*. Estão identificados por hora de início e hora de fim da ocorrência, assim como pela *edge* afetada.

#### 4.2.6 Thresholds

A tabela *Thresholds* contém os 4 patamares relativos ao *traffic status* e respetivos limites entre patamares para cada tipo de via rodoviária. Por exemplo, uma velocidade que poderá ser considerada como lenta numa auto-estrada, pode ser considerada normal ou elevada numa estrada nacional. Em termos de significado, o valor 1 representa trânsito livre, aumentando consoante o agravamento das condições de tráfego, em que o valor 4 representa um nível de trânsito muito congestionado.

#### 4.2.7 Histórico de previsões

As previsões efetuadas pelos algoritmos implementados na Armis ITS estão presentes em tabelas dedicadas a cada método, sendo que para este estudo foram utilizados apenas os dados de previsão dos métodos Rede Neuronal Artificial e Micro-simulação: *HistForecastedANNDATA* e *HistForecastedMicroData*, respetivamente. Estas apresentam para um determinado momento e uma determinada *edge*, as previsões realizadas pelo método de previsão associado à mesma tabela, desde 5 minutos até 2 horas (em intervalos de 5 minutos). Estas previsões contemplam os valores estimados para a velocidade média, volume, ocupação, e um *traffic status* resultante.

#### 4.2.8 Previsões com expert ANN

As primeiras experimentações realizadas pela Armis com o intuito de criar um agente *expert* capaz de selecionar o melhor método de previsão para cada estado da rede recaíram sobre a implementação de um modelo de Rede Neuronal Artificial (ANN). Assim, a tabela *HistForecastedData*

apresenta, para cada estado da rede, a previsão final calculada pelo método escolhido por essa Rede Neuronal Artificial.

#### 4.2.9 Análise descritiva dos dados

Através da análise dos dados disponibilizados, foi possível retirar algumas conclusões descritivas dos mesmos. Durante os primeiros 24 dias do mês de janeiro de 2014, para as 27 *edges* com dados completos, geraram-se 740 282 estados da rede com respectivas previsões, sendo alvo de estudo na presente dissertação.

Relativamente ao *traffic status* observado nos dados históricos, mais de 88% dos intervalos de tempo o tráfego fluía completamente livre contra apenas 1,5% de ocasiões de tráfego muito congestionado (detalhes no gráfico da figura 4.1).

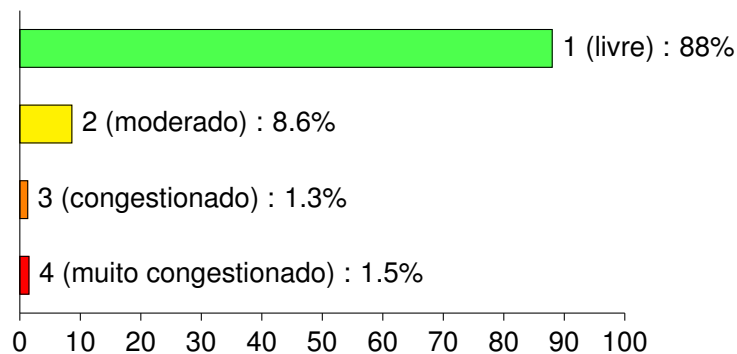


Figura 4.1: Gráfico com proporções de *traffic status* dos estados da rede no *dataset*

Em termos de condições meteorológicas, em 71% das ocasiões encontrava-se tempo seco, enquanto que em 29% chovia.

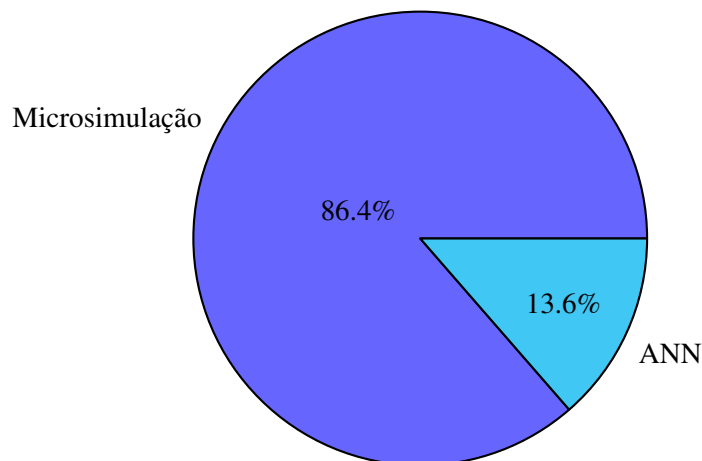


Figura 4.2: Gráfico com proporção de melhor método de previsão para os estados da rede no *dataset*

Com o objetivo de perceber o peso que cada um dos métodos de previsão tem no domínio desta dissertação, utilizaram-se os dados de previsão disponibilizados e os dados sensoriais reais



para determinar, para cada intervalo de previsão, qual o método de previsão com resultado mais próximo da realidade. Verificou-se assim que, das 740 282 previsões, 639 289 delas (86,4%) apresentam a Microsimulação como método com maior precisão, e os restantes 100 953 (13,6%) são previstos com mais exatidão pelo método ANN (gráfico da figura 4.2).

Contudo, existe ainda outro conhecimento que importa extrair dos dados existentes, e que é a quantidade de casos em que cada algoritmo de previsão apresenta melhores resultados, mas agora agrupado por *traffic status*. Esse conhecimento está expresso no gráfico da figura 4.3 em forma de percentagens. Do ponto de vista analítico, observa-se que para um *traffic status* de nível 1 (trânsito livre), 88% dos casos de previsão apresentam melhor performance utilizando Microsimulação. No entanto, à medida que o nível de *traffic status* aumenta, ou seja, aumento do congestionamento do tráfego, o método ANN ganha mais peso. Para um nível 4 (muito congestionado), 35% dos casos de previsão apresentam melhor performance utilizando ANN contra 65% da Microsimulação.

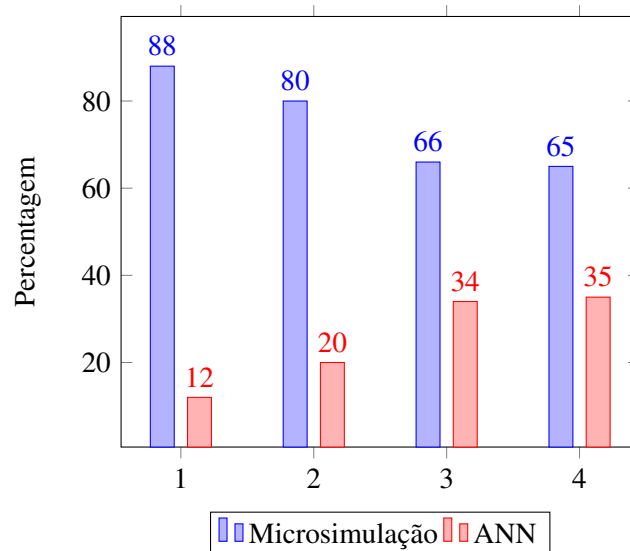


Figura 4.3: Gráfico de proporção de melhor método de previsão, agrupado por *traffic status*.

### 4.3 Perspetivas de solução

A solução proposta para o problema apresentado passa essencialmente pela implementação de um módulo inteligente (também denominado "*expert*") baseado em *machine learning* que seja capaz de, em tempo real, e com base no estado da rede do momento, prever e seleccionar o método de previsão com maior eficácia para esse mesmo estado. Para tal, são utilizados inicialmente dados históricos para realizar a aprendizagem dos modelos implementados.

Através do estudo e revisão da literatura, propôs-se inicialmente quatro soluções distintas: *Q-Learning*, SARSA (State-Action-Reward-State-Action), CBR (Case-based Reasoning) e Árvores de Decisão (algoritmo C4.5). No entanto, o método SARSA não foi considerado como uma solução pelo motivo explicado mais à frente, onde será então detalhada a conceção e detalhe dos passos envolvendo cada um destes métodos.

### 4.3.1 Arquitetura genérica do módulo *Expert*

O módulo inteligente a desenvolver, independentemente do algoritmo preditivo a utilizar, estará dividido em três camadas principais: camada de acesso à base de dados, camada de aprendizagem do modelo, e camada de classificação de novas instâncias. O diagrama da figura 4.4 representa esta arquitetura.

A camada de acesso à base de dados, comum a todas as implementações, permite recuperar todos os dados necessários. É utilizada também no método *Q-Learning* para inserir e atualizar entradas na tabela de valores *Q*.

Na camada de aprendizagem, são utilizados os dados recuperados para realizar a aprendizagem computacional do modelo, nomeadamente os dados históricos de medições sensoriais, informações meteorológicas e de acontecimento de incidentes, e histórico de previsões de tráfego efetuadas.

Por fim, na camada preditiva, são classificadas novas instâncias de estados da rede. Em tempo real, partindo do estado da rede atual, esta camada permite prever qual o método de previsão de tráfego mais apropriado para o esse mesmo estado.

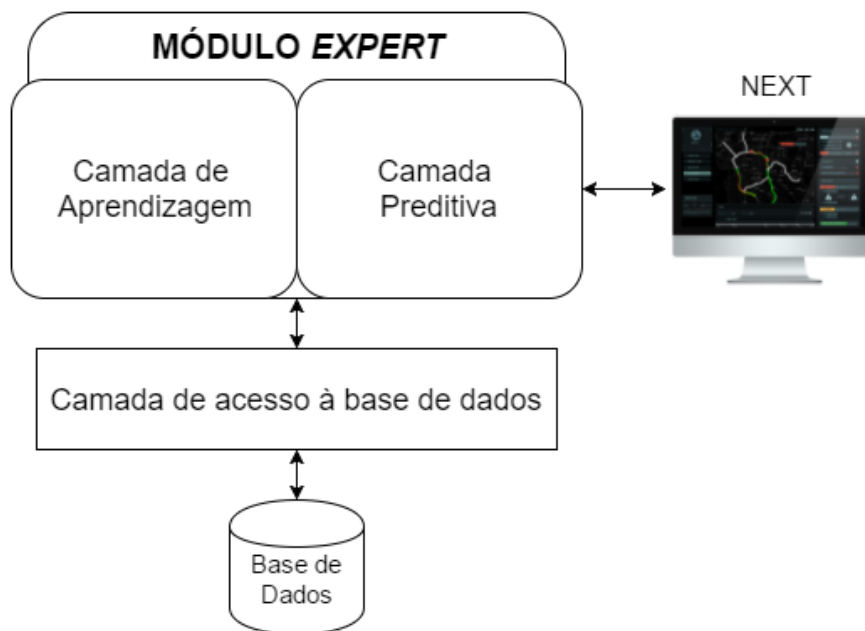


Figura 4.4: Arquitetura do módulo *expert*.

#### 4.3.1.1 Ciclo de Reaprendizagem

Tendo em consideração que poderão existir alterações nos estados da rede, tais como novos e diferentes dados provenientes de sensores, alterações na topologia da rede rodoviária ou alterações na sinalização, é fundamental que existam ciclos de reaprendizagem. Assim, garante-se que os algoritmos de decisão se encontram adequados ao estado atual da rede rodoviária.

### 4.3.2 *Q-Learning*

O primeiro método de aprendizagem abordado é o algoritmo de aprendizagem por reforço *Q-Learning* [WD92]. Como descrito na revisão da literatura, neste algoritmo o agente tem como objetivo aprender uma política ótima através da sua interação com o meio, que se traduz em sequências estado-ação e respetiva recompensa. Ou seja, encontrando-se o agente num estado  $s$  e realizando uma ação  $a$ , recebe um *reward* (recompensa)  $r$  e transita para o estado  $s'$ .

Transpondo a teoria para o domínio do nosso problema, o estado  $s$  corresponde ao estado da rede atual e sobre o qual se pretende gerar previsões, a ação  $a$  corresponde ao método de previsão a ser selecionado e o estado final  $s'$  corresponde à previsão obtida por esse mesmo método. Consequentemente, o *reward*  $r$  corresponde à recompensa por essa escolha, baseado na proximidade entre a previsão efetuada por esse método e a realidade, obtida posteriormente através dos dados sensoriais.

Podemos considerar este como um problema de minimização, pois quanto menor for a diferença entre o estado previsto e a realidade, ou seja, o erro, melhor será a previsão. Nesse sentido, a política do agente incide na escolha da ação com menor valor  $Q$ , ao contrário do que é mais habitual. Outro exemplo semelhante onde a política prevê a escolha da ação com menor valor  $Q$  é no caso da escolha do algoritmo mais eficiente com base no seu tempo de execução [LL00], que interessa ser minimizado.

Como introduzido na revisão de conceitos, e tendo em conta o problema de minimização, a equação geral de atualização do valor  $Q$  é representada por:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[ R_{t+1} + \gamma \min_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \quad (4.1)$$

onde  $s_t$  representa o estado num momento  $t$ ,  $a_t$  representa a ação tomada no instante  $t$ ,  $R_{t+1}$  é o custo por essa decisão, e  $\alpha$  é a taxa de aprendizagem.

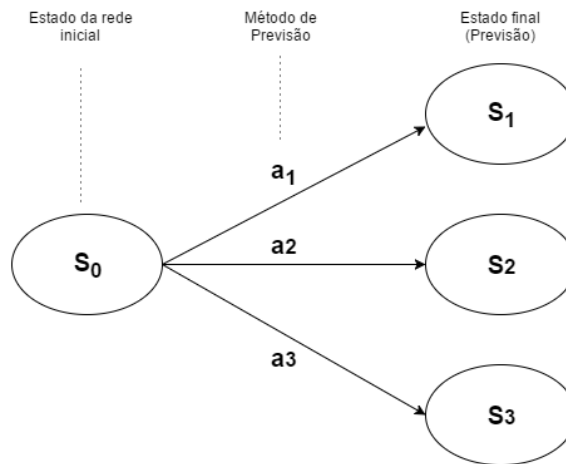


Figura 4.5: Diagrama do algoritmo *Q-Learning*.

No entanto, no domínio atual, o algoritmo está restringido a apenas uma transição ou passo

como se pode observar no diagrama da figura 4.5, onde  $S_0$  representa o estado inicial (estado da rede), e  $S_1$ ,  $S_2$  e  $S_3$ , os estados finais (previsões) que se podem obter através das ações (métodos de previsão)  $a_1$ ,  $a_2$  e  $a_3$ , respetivamente. Assim, o estado resultante  $s_{t+1}$  é um estado final e tem um custo de zero. Desta forma, a equação de atualização pode ser reduzida a:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [R_{t+1} - Q(s_t, a_t)] \quad (4.2)$$

onde, para calcular o novo valor  $Q$ , apenas temos que ter em consideração o valor  $Q$  atual para o par estado-ação e o *reward* obtido nesta transição, devidamente multiplicado pelo fator de aprendizagem.

#### 4.3.2.1 Fator de aprendizagem - $\alpha$

O fator de aprendizagem é responsável por indicar a percentagem de *reward* que irá influenciar a atualização do valor  $Q$ . Assim, caso fosse 0, o agente não aprenderia, e caso fosse 1 o agente apenas consideraria o *reward* mais recente. Embora o fator ideal seja aquele que permita uma convergência mais rápida dos valores  $Q$  de todos os pares estado-ação, há uma constante utilizada frequentemente, tal que  $\alpha_t(s, a) = 0.1$  para qualquer  $t$ . No entanto, no âmbito desta implementação, devido ao *dataset* diminuto que foi disponibilizado e que origina poucas iterações de aprendizagem para a maioria dos pares estado-ação, considerou-se um fator de 0.2, permitindo acelerar um pouco o processo de aprendizagem.

#### 4.3.2.2 Condição inicial $Q_0$

Sendo o *Q-Learning* um algoritmo iterativo, há que definir um valor  $Q$  inicial a aplicar a todos os pares estado-ação. Quando um determinado estado a ser calculado já se encontra presente na tabela  $Q$ , é suficiente a sua atualização baseada na fórmula do algoritmo. No entanto, quando se trata da primeira entrada, é necessário decidir qual o valor a adotar inicialmente. Na literatura são apresentadas 3 opções, principalmente: iniciar com o valor zero, com o valor máximo ou relativamente grande, ou com o primeiro valor de *reward*. Como já referido, o valor de reforço atribuído pela função de atualização definida para este problema representa o valor de erro entre a previsão e a realidade. Assim, e tendo em conta que não é um valor que pode ser normalizado, a segunda opção foi descartada. Por fim, para evitar grandes desvios iniciais causados por um eventual erro elevado único, assumiu-se o valor zero como condição inicial para todos os pares estado-ação:  $Q_0 = 0$ .

#### 4.3.2.3 Tabela $Q$ e caracterização dos estados

O algoritmo *Q-Learning* utiliza uma tabela para armazenar os seus pares estado-ação. A primeira tarefa é definir um espaço de estados da rede finito, caracterizados por um conjunto de atributos dispostos em intervalos discretos. É necessário, porém, encontrar um equilíbrio nesses atributos, que permita agrupar os estados bastante idênticos, sem individualizar em demasia e levar

a um espaço de estados muito elevado e com uma aprendizagem inferior. Assim, e partindo dos dados fornecidos e anteriormente descritos, foram selecionados e adaptados os seguintes atributos para identificar cada estado da rede:

- *EdgeID*, identificador da *edge* ou troço de uma via;
- *Traffic Status*, que pode tomar valores de 1 (livre) a 4 (muito congestionado);
- Tipo de dia, que pode ser: dia da semana, fim-de-semana, feriado, véspera de feriado ou "ponte"(sendo que os 3 últimos se sobrepõe aos 2 primeiros);
- Condições Atmosféricas (chuva ou tempo seco)
- Período de Férias (sim ou não);
- Ocorrência de incidentes (sim ou não)
- Intervalo de previsão, de 5 minutos até 2 horas, em intervalos de 5 minutos (assumindo que uma previsão a 5 minutos poderá ser bastante diferente de uma previsão a 2 horas);
- Altura do dia, identificada em 48 intervalos de 30 minutos.

Para cada estado, caracterizado por estes 8 atributos discretos, existe uma entrada para cada método de previsão e respetivo valor  $Q$ .

Após uma primeira fase de treino com base nos dados históricos, o algoritmo terá a capacidade de prever qual o melhor método de previsão para um novo estado da rede. Para tal, após discretização dos atributos do novo estado, estes são utilizados para recuperar os pares estado-ação correspondentes e respetivos valores  $Q$ . Agora, seguida a política definida, é selecionada a ação com menor valor  $Q$ , que corresponde ao método de previsão com menor erro para o estado da rede considerado. Por fim, o valor  $Q$  é atualizado considerando o novo *reward* obtido, calculado a partir do erro entre a previsão e a realidade. Como forma de tirar partido de todos os dados disponibilizados, e dado que estamos perante um algoritmo *Q-Learning* de apenas uma transição e com um curto espaço de ações (algoritmos de previsão), a fase de *exploration* irá abordar as transições para todos os algoritmos e utilizar os respetivos reforços para atualização do seu valor  $Q$ .

No final, teremos pouco mais de 184 mil estados da rede distintos para cada *edge*. Multiplicado pelo número de métodos de previsão (ou ações) e pelo número de *edges* disponíveis, teremos o número máximo de entradas da tabela. No presente, para 2 métodos e 27 *edges*, a complexidade máxima é perto de 10 milhões de entradas (9 953 280).

Ainda assim, foram tomadas algumas medidas com o intuito de diminuir a quantidade de pares estado-ação. Apesar dos dados disponibilizados se encontrarem divididos em intervalos de 5 minutos, o que totalizaria 288 intervalos por dia, estes foram agrupados em intervalos de 30 minutos, totalizando assim apenas 48. Para além do benefício espacial, admitiu-se que em intervalos de 30 minutos a diferença de tráfego pode ser desprezada para já, permitindo obter também mais casos de aprendizagem em cada par estado-ação. Outra medida para reduzir o espaço de casos foi a junção num único atributo do tipo de dia. Assim, o mesmo indicará, assumindo esta ordem de

prioridade, se é um feriado, "ponte", véspera de feriado, e dia da semana ou de fim-de-semana se não for nenhum dos anteriores.

A aprendizagem realizada pelo *Q-Learning* decorre em duas fases distintas: *exploration*, onde serão testados todos os casos possíveis e utilizar o seu reforço para atualização dos valores da tabela  $Q$ , e *exploitation*, tipicamente utilizada em tempo real e onde é selecionada uma ação (método de previsão) seguindo a política  $Q$  definida e o reforço é utilizado para atualizar o valor  $Q$  apenas desse par estado-ação.

#### 4.3.2.4 Fase *Exploration*

Numa primeira fase, a aprendizagem será realizada com base nos dados históricos recolhidos de anos anteriores. Assim, para cada estado da rede, num dado momento, o algoritmo *Q-Learning* irá realizar tantas transições quantos métodos de previsão o sistema suportar, aplicando o *reward* obtido nas respetivas entradas da tabela  $Q$ .

Embora, conceptualmente, na fase de *exploration* seja escolhida uma ação aleatória para cada estado, no caso em estudo serão exploradas todas as ações possíveis, dado que existem dados já calculados anteriormente para os vários algoritmos de previsão e o número de transições é limitado ao número de métodos de previsão. Assim, é possível realizar uma aprendizagem mais completa, embora com mais esforço computacional.

#### 4.3.2.5 Fase *Exploitation*

A fase de *exploitation* consiste numa aprendizagem *on-line*, ou seja, durante a fase de execução em tempo real do sistema de previsão de tráfego. Nesta fase, para um novo estado da rede, a transição a aplicar deriva de uma política gananciosa que consiste na seleção da ação (método de previsão) com melhor valor  $Q$ , neste caso com o valor  $Q$  inferior, entre todas as transições possíveis. Será então este método de previsão o executado pelo sistema e o reforço obtido será utilizado na atualização do valor  $Q$  para o par estado-ação selecionado.

### 4.3.3 SARSA

O segundo método de aprendizagem considerado foi o SARSA (*State-Action-Reward-State-Action*) [RN94] que, tal como o *Q-Learning*, é um método de aprendizagem por reforço. Observando novamente as etapas do algoritmo *Q-Learning*, o agente começa no estado  $s_1$ , pratica uma ação  $a_1$  que o conduz para o estado  $s_2$ , recebendo um *reward*  $r_1$ , e considera a máxima recompensa possível para uma ação a partir do estado  $s_2$ , usando esse valor para atualizar o valor da ação  $a_1$  sobre o estado  $s_1$ . No algoritmo SARSA, o agente também começa no estado  $s_1$ , pratica a ação  $a_1$  e recebe o *reward*  $r_1$  mas, agora no estado  $s_2$ , realiza outra ação ( $a_2$ ) e recebe um *reward* do novo estado antes de voltar atrás e atualizar o valor da ação  $a_1$  sobre o estado  $s_1$ .

Ou seja, o algoritmo SARSA realiza mais uma ação sobre o novo estado e é o *reward* dessa ação que utiliza para a atualização, ao contrário do *Q-Learning* que utiliza o valor  $Q$  máximo

conhecido para as transições possíveis. Por fim, o valor  $Q$  para o par estado-ação é atualizado segundo a equação:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (4.3)$$

que é muito semelhante à equação do algoritmo *Q-Learning*. Agora, aplicando o algoritmo ao domínio desta dissertação percebemos que, à semelhança do que acontece no *Q-Learning*, apenas existe uma transição entre o estado inicial (estado da rede inicial) através de uma ação (método de previsão) atingindo um novo estado (previsão futura do estado da rede), que é um estado final. Desta forma, e analisando a fórmula do algoritmo SARSA, percebemos que a componente respeitante ao valor  $Q$  resultante da ação  $a_2$  sobre o estado  $s_2$  será zero. Assim, a equação simplificada será:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [R_{t+1} - Q(s_t, a_t)] \quad (4.4)$$

que, como se pode observar, torna-se exatamente igual à equação simplificada, para este domínio, do algoritmo *Q-Learning*, pois a componente que distingue estes dois algoritmos é anulada.

Após esta análise, pode-se concluir que não existe utilidade na implementação deste algoritmo pois devido à adaptação ao presente domínio torna-se exatamente igual ao algoritmo *Q-Learning*, cuja abordagem foi previamente descrita.

#### 4.3.4 CBR

*Case-based Reasoning*, é uma metodologia que permite tomar decisões para casos novos com base no conhecimento adquirido na resolução de problemas anteriores [Kol92, AP94]. No presente problema, espera-se que o CBR permita obter bons resultados na medida em que, através dos dados históricos que se possui, poderá ser construída uma base de conhecimento (denominada *case base* no contexto do CBR) que permita inferir qual o método de previsão mais apropriado para novos casos. O *case base* é assim o componente principal de um sistema CBR.

Como é descrito por Watson [Wat99], CBR não se trata de um algoritmo ou tecnologia, mas sim de uma metodologia, assente em 4 etapas: recuperação de casos semelhantes, reutilização de soluções passadas para novos casos, revisão e adaptação da solução (se necessário), e retenção do novo caso e respetiva solução na base de conhecimento. O maior desafio na implementação desta metodologia passa pela decisão do algoritmo a utilizar para a indexação de novos casos e recuperação de casos semelhantes.

Após revisão da literatura sobre aplicação em casos semelhantes, optou-se pela conceção do sistema CBR com recurso ao algoritmo *k-Nearest Neighbors* (k-NN). Segundo Watson [Wat99], esta técnica deverá ser a mais utilizada pois é a fornecida pela maioria das ferramentas CBR.

#### 4.3.4.1 Case base, indexação e recuperação de casos

O *case base* consiste então no conjunto de instâncias criadas a partir dos dados históricos e o algoritmo com menor erro de previsão para cada uma delas, atributo de classificação. Assim sendo, cada instância representa um ponto num espaço de  $n$  dimensões, sendo  $n$  o número de atributos que caracterizam uma instância.

Dado que não existe nenhuma fórmula de semelhança entre *edges* que nos permita inferir comportamentos de tráfego idênticos em diferentes *edges*, aliado ao facto de existirem poucas *edges* com dados históricos completos atualmente, decidiu-se individualizar as *edges*. Desta forma, teremos uma instância de aprendizagem individual para cada *edge*, seguindo o mesmo modelo com os mesmos atributos. Por outras palavras, teremos um *case base*, implementado pelo algoritmo k-NN, para cada *edge*.

Após análise do *dataset*, foram então seleccionados os seguintes 10 atributos:

- Altura do dia (em intervalos de 5 minutos)
- Intervalo de previsão (até 2 horas, em intervalos de 5 minutos)
- *Traffic Status* (1 a 4)
- Condições atmosféricas (chuva ou tempo seco)
- Incidente (sim ou não)
- Tipo de dia (dia da semana ou fim-de-semana)
- Período de Férias (sim ou não)
- Feriado (sim ou não)
- Véspera de feriado (sim ou não)
- Dia de "Ponte"(sim ou não)

Dado que todos estes atributos podem ser transformados em valores numéricos normalizados, optou-se pela função da distância Euclidiana para identificação dos "vizinhos" mais próximos. Assim, a similaridade entre dois casos pode ser representada pela função:

$$Similaridade(A, B) = \sum_{i=1}^n f(A_i, B_i) \times w_i \quad (4.5)$$

em que  $A$  representa o novo caso,  $B$  representa um caso na base de conhecimento,  $n$  é o número de atributos (10 neste caso),  $f(A_i, B_i)$  representa a função da distância Euclidiana do atributo  $i$  entre os casos  $A$  e  $B$ , e  $w_i$  representa o peso do atributo  $i$  para o cálculo que, por predefinição, será considerado igual para todos os atributos, ou seja,  $1/10 = 0.1$ .



#### 4.3.4.2 Parametrização

Relativamente ao valor  $k$  ideal, deseja-se o teste com o 1, 3, 5 e 7 vizinhos, a fim de descobrir qual o valor que permite obter uma taxa de acerto mais elevada, tendo também em conta que o número de casos de treino disponíveis poderá condicionar este valor. No caso da procura por apenas 1 vizinho, o novo caso obterá a mesma classificação deste. Para um  $k$  superior a 1, será atribuída a classificação predominante, ou seja, o método de previsão utilizado na maioria dos vizinhos encontrados.

#### 4.3.4.3 Etapas

Sumariando, o método CBR a aplicar neste projeto consiste nas seguintes etapas:

1. Criação do *case base*, através de treino do modelo k-NN utilizando os dados históricos disponíveis;
2. Perante um novo caso, obtenção dos  $k$  vizinhos mais próximos;
3. Classificação do novo caso mediante a classificação dos vizinhos;
4. Validação da classificação do novo caso após chegada dos novos dados históricos e inserção do mesmo, classificado corretamente, na base de conhecimento (*case base*).

#### 4.3.5 Árvore de Decisão

De forma intuitiva, e tal como observado no método anterior, a seleção de algoritmos é um problema de classificação, ou seja, pretende-se categorizar um novo estado da rede segundo o método de previsão que deverá ser mais apropriado. Assim sendo, decidiu-se utilizar também o método de Árvore de Decisão para realizar esta tarefa preditiva, por ser uma técnica amplamente aceite e por permitir obter uma representação visual mais intuitiva ao Ser Humano. A criação desta árvore ficará a cargo do algoritmo C4.5, utilizando como *dataset* o conjunto de dados disponibilizado pela Armis e descrito anteriormente. Desta forma, pretende-se treinar a árvore de decisão com base em entradas que incluem os atributos do estado da rede num dado instante e uma classificação, que identificará o método de previsão com uma previsão mais próxima da realidade para esse caso.

A caracterização de cada caso será feita pelo mesmo conjunto de 10 atributos utilizado no método CBR.

Tal como o método CBR, e pelo mesmo motivo indicado, nas condições atuais pretende-se criar uma árvore de decisão para cada *edge* disponível.

Em resumo, quando em execução para uma determinada *edge*, o agente munido desta árvore de decisão terá a capacidade de, dada um novo estado da rede, identificar qual o método de previsão de tráfego mais adequado para o mesmo.

#### 4.4 Cálculo do erro em previsões

Como se pode observar pelos dados disponibilizados pela Armis ITS, o estado do tráfego num dado momento pode ser classificado através do *traffic status*. Este classificador, que assume valores entre 1 (trânsito livre) e 4 (trânsito muito congestionado), deveria ser calculado através de uma fórmula que englobasse as três métricas caracterizadoras do fluxo de tráfego: volume de veículos, velocidade média e ocupação do sensor. No entanto, neste momento o *traffic status* ainda tem apenas em consideração a velocidade média obtida dos dados sensoriais tendo em conta limites (habitualmente chamados *thresholds*) definidos para o tipo de via onde se encontra. Uma das etapas mais importantes dos métodos supra-citados passa por descobrir a distância (ou erro) entre o estado do tráfego previsto e o estado real, obtido normalmente 5 minutos após a previsão em tempo-real, com o recebimento de novos dados dos sensores. Tendo em conta que o *traffic status* é algo ainda muito pouco explorado e que se baseia apenas na velocidade média do tráfego, o cálculo do erro implementado nos métodos utilizados nesta dissertação é o módulo da diferença entre a velocidade da previsão e a velocidade real, para o mesmo instante:

$$Erro(P_t, R_t) = |V_{P_t} - V_{R_t}| \quad (4.6)$$

sendo  $P_t$  a previsão para o instante  $t$ ,  $R_t$  o estado da rede real lido pelo sensor para o instante  $t$ , e  $V$  a velocidade contida num determinado estado da rede.

Esta fórmula de cálculo do erro permite, em CBR e Árvore de Decisão, determinar qual o método de previsão com um resultado mais próximo da realidade num determinado estado da rede que será o atributo de classificação. No caso do *Q-Learning*, o valor do erro é utilizado como *reward* na aprendizagem por reforço efetuada.

#### 4.5 Metodologia de teste e análise de resultados

Após a implementação do método de aprendizagem por reforço e dos métodos preditivos, torna-se importante realizar uma análise e avaliação de desempenho com base em resultados obtidos após a fase de aprendizagem.

Todos os métodos propostos são, direta ou indiretamente, métodos de predição ou classificação, indicando, para um determinado estado da rede, o método de previsão que deverá oferecer melhores resultados. Assim, e visto que se trata de uma classificação binária atualmente (Microsimulação ou ANN), a avaliação destes métodos será realizada através da análise de matrizes de confusão geradas a partir dos resultados de acerto e falha na sua previsão do método com uma previsão mais próxima da realidade. Pretende-se assim calcular a precisão (*precision*) e sensibilidade (*recall*) das classificações.

Contudo, não esquecendo que o objetivo final do sistema de previsão de tráfego é maximizar o acerto no *traffic status* das suas previsões, pretende-se também verificar qual o contributo final destes módulos para o sistema final. Para isso, pretende-se realizar outro teste em que se terá apenas em conta se o algoritmo escolhido apresenta uma previsão correta. Este teste permite

despistar casos em que, embora o *expert* tenha selecionado um método de previsão que gera uma previsão muito próxima da realidade mas que não seria o melhor método e, pela avaliação realizada anteriormente, o método *expert* é penalizado.

Os testes serão adaptados a cada método *expert*. Para o *Q-Learning*, é realizada uma fase de *exploration*, onde se pretende que os valores *Q* correspondentes aos pares estado-ação converjam para um valor ótimo. Posteriormente, novos casos serão testados, obtendo um método de previsão a partir da política que indicará a ação com menor valor *Q*. O método CBR, que é suportado por um modelo k-NN, e as Árvores de Decisão, tirarão partido da técnica de *cross-validation* para obtenção de resultados e consequente validação.

A demonstração dos resultados será feita com recurso a técnicas estatísticas. Como se trata de um problema de classificação entre dois métodos, decidiu-se utilizar a matriz de confusão para demonstrar a precisão e sensibilidade das classificações em cada caso de teste.

		Classe Real		
		Microsimulação	ANN	<i>Class Precision</i>
Classe Prevista	Microsimulação	A	B	E
	ANN	C	D	F
	<i>Class Recall</i>	G	H	

Tabela 4.1: Estrutura da matriz de confusão.

Na tabela 4.1 podemos observar a estrutura da matriz de confusão para o problema em estudo. As classes das linhas representam as previsões do *expert* e as classes das colunas representam os métodos de previsão de tráfego com maior precisão após experimentação. As células marcadas com letras têm os seguintes significados:

- A é o número de previsões corretas de Microsimulação;
- B é o número de previsões erradas de Microsimulação;
- C é o número de previsões corretas de ANN;
- D é o número de previsões erradas de ANN;
- E é a percentagem de acerto na previsão Microsimulação;
- F é a percentagem de acerto na previsão ANN;
- G é a percentagem de casos de Microsimulação corretamente classificados;
- H é a percentagem de casos de ANN corretamente classificados.

## 4.6 Sumário

Este capítulo pode ser classificado como componente central nesta dissertação por definir concretamente o problema e discutir as soluções propostas para a sua resolução. Foram apresentadas as tarefas e decisões que servem de base à implementação do módulo *expert*.

Inicialmente foi realizado um enquadramento global dos dados disponibilizados e que são essenciais ao desenvolvimento desta dissertação, sendo posteriormente adaptados a cada método na fase de implementação.

Na secção 4.3, parte mais importante deste capítulo, são discutidas as abordagens utilizada na elaboração de soluções para o presente problema, nomeadamente com recurso aos métodos *Q-Learning*, SARSA (que foi fundamentadamente descartado), *Case-based reasoning* e Árvore de Decisão. Relativamente ao *Q-Learning* salienta-se a explicação que indica que as condições de atuação no domínio em estudo justificam a redução feita à equação global de atualização de valores  $Q$ . Relativamente à metodologia CBR são apresentados os detalhes das etapas do seu ciclo e é fundamentada a utilização do algoritmo k-NN para indexação e recuperação de novos casos. Relativamente às Árvores de Decisão, são apresentados os pormenores da solução e seu enquadramento.

O capítulo termina com a explicação da fórmula utilizada para o cálculo do erro cometido nas previsões e com a metodologia de teste e avaliação dos métodos propostos para o agente *expert*.

## Capítulo 5

# Implementação

Este capítulo apresenta, de forma detalhada, os passos e mais algumas decisões tomadas durante o processo de implementação dos agentes "*experts*" baseados nos métodos de *machine learning* abordados no capítulo anterior: *Q-Learning*, CBR e Árvore de Decisão.

### 5.1 Ferramentas utilizadas

Na implementação das soluções de aprendizagem computacional propostas, foi utilizado o Eclipse Java IDE para o desenvolvimento do código fonte dos agentes "*experts*", organizado pela ferramenta de gestão de projetos Maven.

A base de dados onde se encontram as tabelas utilizadas para aprendizagem dos modelos, assim como todos os restantes dados necessários ao desenvolvimento, encontra-se no sistema de gestão de base de dados Microsoft SQL Server 2016.

Foi ainda utilizada a ferramenta RapidMiner Studio para mais rapidamente testar os algoritmos e suas respetivas parametrizações, e realizar validação através de técnicas de treino e teste de performance dos modelos. Inicialmente foi também útil para realizar uma análise descritiva dos dados a serem utilizados.

### 5.2 Acesso a dados

A camada comum de acesso aos dados essenciais foi implementada também em linguagem Java, utilizando o JDBC Driver para MS SQL Server. Consequentemente, foram também implementadas e encapsuladas todas as *queries* para obtenção dos dados necessários.

### 5.3 Módulo *expert*

O módulo *expert*, transversal a todos os métodos implementados, incorpora a camada de acesso a dados, a camada de aprendizagem, onde são utilizados os dados históricos para aprendizagem do modelo respetivo, e a camada preditiva onde, para um novo estado da rede, é indicado o

método de previsão adequado.

## 5.4 *Q-Learning*

Como descrito no capítulo 4, o primeiro método de aprendizagem a ser implementado foi o *Q-Learning*. Foi feita uma implementação de raiz, adequada ao problema em questão, utilizando a camada comum de acesso à base de dados. O primeiro passo diz respeito à criação de uma nova tabela na base de dados que corresponderá à tabela *Q*, onde são armazenados os pares estado-ação e respectivos valores de aprendizagem *Q* (ver tabela 5.1). De seguida foi implementado o módulo de aprendizagem com base nos dados históricos, correspondendo à fase de *exploration*. Esta fase recebe como *input* uma data/hora de início de data/hora de fim, assim como um conjunto de *edges* que idealmente conterão dados completos para o mesmo intervalo. Assim, o processo de aprendizagem passa por iterar sobre intervalos de 5 minutos desde a data inicial até à data final.

Estado da rede	Algoritmo	Q
A	1	$Q_{A1}$
A	2	$Q_{A2}$
B	1	$Q_{B1}$
B	2	$Q_{B2}$
...	...	...

Tabela 5.1: Estrutura da tabela *Q*.

Para cada uma dessas iterações, que corresponde a um momento em que são efetuadas previsões para as 2 horas seguintes, são realizadas para cada *edge* as seguintes tarefas para caracterização do estado da rede nesse momento e nessa *edge*:

1. Obtenção e tratamento dos dados caracterizadores do dia correspondente, nomeadamente dia da semana, se é um feriado, véspera de feriado ou "ponte", e se pertence a um período de férias escolares;
2. Obtenção das métricas correspondentes às condições de tráfego no momento e *edge* em questão: velocidade média, volume de veículos, ocupação do sensor e *traffic status*;
3. Obtenção do identificador das condições atmosféricas mais recentes, presentes em intervalos de 30 minutos, e que neste momento apenas contemplam os estados "limpo" e "chuva";
4. Procura por informação de algum incidente com início anterior e fim posterior à data e hora da previsão atual.

De seguida, são processadas todas as previsões realizadas para esse momento, que vão desde os cinco minutos seguintes até duas horas, em intervalos de cinco minutos, totalizando 24 previsões. Juntando o valor do intervalo de previsão às características enumeradas anteriormente, obtém-se um estado no domínio do algoritmo *Q-Learning*.

Como forma de tirar todo o partido na fase de *exploration*, são realizadas as duas ações possíveis para cada estado, que correspondem aos dois algoritmos disponíveis atualmente: ANN e Microsimulação. Destas duas transições são calculados os respetivos *rewards* com base no erro da previsão. Para tal, são efetuados dois passos:

1. Obtenção dos valores resultantes da previsão efetuada por cada um dos algoritmos, e que se encontra nas tabelas *HistForecastedMicroData* e *HistForecastedANNData*: volume, ocupação, velocidade média e *traffic status*;
2. Obtenção dos valores reais para as quatro métricas na data/hora e *edge* previstas, contidos na tabela de dados históricos de leituras dos sensores: *HistData*.

Com estes valores disponíveis, é calculada a diferença (erro) entre o estado da rede previsto pela Microsimulação e o estado da rede real, assim como a diferença entre o estado da rede previsto pela Rede Neuronal (ANN) e o estado da rede real. Como discutido anteriormente, neste momento esta diferença representa a diferença entre as velocidades previstas e a velocidade real medida pelos sensores. Com estas diferenças, temos por fim o reforço a utilizar para os dois pares estado-ação processados.

O último passo de cada iteração passa pela atualização da tabela  $Q$ . Para tal, é procurado na tabela o par estado-ação em causa, através de uma pesquisa por todas as características do estado da rede no momento, intervalo de previsão e algoritmo de previsão (ação). Caso este par já exista na tabela, é efetuada a atualização do valor  $Q$ , senão é inserido o novo par com o respetivo valor  $Q$ .

No final de várias iterações, no caso ideal teremos o valor  $Q$  dos pares estado-ação a convergir para valores que traduzem a precisão do respetivo algoritmo na previsão para um estado da rede com as características presentes nesse mesmo estado.

## 5.5 CBR

Outro método desafiante a ser implementado foi o *Case-based Reasoning* (CBR). Como descrito anteriormente, pretende-se que, com base em casos anteriores e seus resultados, seja possível aferir qual a melhor solução para um novo caso. Desta forma, desenvolveu-se um módulo que representa este método e implementa o ciclo CBR. Inicialmente, tirando partido novamente da base de dados de estados da rede reais e previsões efetuadas por Microsimulação e ANN, foi realizada a etapa de aquisição de conhecimento que se traduz no preenchimento da *case base* com as previsões já existentes. Sendo cada caso composto por um conjunto de atributos que caracterizam o estado da rede e o intervalo de previsão, a sua solução é representada pelo algoritmo com melhor precisão na previsão, nesse caso concreto.

Prosseguindo com pormenores relativos à implementação, a parte fulcral deste método passava pelo desenvolvimento das etapas de indexação e de recuperação de casos semelhantes. Após revisão da literatura, e como já detalhado no capítulo anterior, a solução encontrada passa pela utilização do algoritmo k-NN para realização destas tarefas. Para tal foi utilizada a *framework*

*open-source* Weka [HFH<sup>+</sup>09], que disponibiliza uma implementação deste algoritmo através da classe *IBk*.

### Modelo do caso

Foi então criado um modelo do caso, que contempla os atributos identificados no capítulo anterior, todos em formato numérico: *traffic status* no momento, período do dia em intervalos de 5 minutos, intervalo de previsão, identificador de condições atmosféricas (tempo seco ou chuvoso), indicador de existência de incidente na via, dia da semana ou fim-de-semana, se é dia de período de férias escolares e se é feriado, véspera de feriado ou "ponte". Estes atributos podem também ser observados na tabela 5.2. Como estes atributos são numéricos, ou foram transformados em tal de forma simples, é utilizada a função distância euclidiana para seleção dos elementos mais próximos, os "vizinhos". A estes atributos é adicionado um classificador, que representa o melhor método de previsão para o estado da rede processado, após verificação das previsões e dados reais.

### Fase de aprendizagem

Inicialmente, para a fase de aprendizagem, é definido um intervalo de datas sobre os quais esta será realizada. Assim, iterando de 5 em 5 minutos, é recuperado o estado da rede no momento da tabela *HistData*, assim como informações do dia em questão, condições meteorológicas e existência de incidentes. Posteriormente, são recuperadas das tabelas *HistForecastedANNData* e *HistForecastedMicrodata* as previsões entre os 5 minutos e as 2 horas posteriores para os métodos ANN e Microsimulação, respetivamente. Com base nessas previsões, e recuperando da tabela *HistData* o estado da rede no período previsto, é calculado qual o método com maior precisão, ou seja, com um erro inferior. Como referido anteriormente, o erro é calculado pelo módulo da diferença entre a velocidade de circulação prevista e a velocidade real medida. Outro aspeto importante prende-se com o facto de todos os atributos, sendo numéricos e discretos, são normalizados segundo a técnica *Min-Max* (valor mínimo e valor máximo).

### Persistência do Modelo

Após a fase de aprendizagem, interessa persistir o modelo de aprendizagem treinado, para que seja possível recuperá-lo posteriormente. Tal é possível tirando partido da classe *SerializationHelper* do package *weka.core*, pertencente à mesma *framework*. Dada a independência existente entre *edges*, como já referido, cada *edge* possui um modelo próprio, sendo que iremos persistir um número de modelos igual ao número de *edges* existente. Posteriormente, com a disponibilização de mais dados e uma análise criteriosa aos mesmos, poderá fazer sentido agrupar por tipo de *edges*, indicado nos dados pelo atributo *EdgeClassID*.



### Utilização em tempo real

Por fim, em contexto real de funcionamento na plataforma de previsão, com a chegada de um novo estado da rede é utilizado o modelo treinado para realizar a classificação desse estado, ou seja, obter a previsão do melhor método de previsão de tráfego para esse estado da rede. Como, por princípio, não serão executados todos os métodos de previsão mas apenas o previsto como sendo melhor pelo sistema CBR, não é possível acrescentar conhecimento ao modelo pois não temos informação se o método de previsão selecionado seria o melhor ou não, mesmo tendo este estado próximo do resultado real ou não. Assim, após algum tempo de execução, deverá ser realizada uma reaprendizagem do modelo com mais casos de treino e, possivelmente, mais recentes.

### 5.6 Árvore de Decisão

O último método *expert* implementado foi uma Árvore de Decisão ou, sendo mais rigoroso, uma árvore de decisão dedicada para cada *edge*. A implementação foi realizada tirando partido da implementação Java J48 do algoritmo C4.5 incluída na *framework* de *data mining* Weka. Como discutido na implementação do método CBR, também para este método faz sentido a criação de uma árvore distinta para cada *edge*, pois as mesmas poderão apresentar comportamentos completamente distintos e que a árvore de decisão não teria forma de avaliar caso um dos atributos considerados fosse o identificador da *edge*. Após ser criado o modelo a partir dos atributos considerados (tabela 5.2), é realizada a fase de aprendizagem. Esta consiste na utilização dos dados caracterizadores recuperados da base de dados para a criação de instâncias de estados da rede. Estes contêm ainda uma *label* de classificação que corresponde ao melhor método de previsão para esse caso. Após a inserção de todas essas instâncias é gerada finalmente a árvore de decisão.

### Parametrização

Em termos de parametrização, é necessário definir o número máximo de níveis que a árvore pode ter, o critério de divisão, e se é desejado realizar “poda” da árvore e com que grau de confiança.

Foram testados vários valores até se concluir qual a combinação com melhores resultados de precisão e que está descrita na secção de resultados.

### Persistência do modelo e ciclo de reaprendizagem

Tal como para o k-NN, também foi necessário persistir as árvores de decisão para que possam ser utilizadas sempre que necessário. Da mesma forma utilizou-se a classe *SerializationHelper* do package *weka.core* para esse efeito.

A reaprendizagem deste modelo tem que ser realizada de forma integral pois não é possível adicionar novas entradas. Assim, para um dado intervalo que deverá ser definido, serão geradas novas árvores com dados mais atualizados.

## Implementação

Atributo	Descrição	Intervalo de valores
<i>Traffic status</i>	<i>Traffic status</i> no momento	1 a 4
<i>5min Slot</i>	Período do dia em intervalos de 5 minutos	1 a 288
<i>Prediction Slot</i>	Intervalo de previsão	1 a 24 (5 minutos a 2 horas)
<i>Day Type</i>	Dia de semana ou de fim-de-semana	0 (semana) ou 1 (fim-de-semana)
<i>Rain</i>	Indicador de Chuva ou tempo seco	0 (tempo seco) ou 1 (chuva)
<i>Incident</i>	Indicador de ocorrência de incidente	0 (não) ou 1 (sim)
<i>Vacation</i>	Indicador de período de férias escolares	0 (não) ou 1 (sim)
<i>Holiday</i>	Indicador de feriado	0 (não) ou 1 (sim)
<i>Holiday's Eve</i>	Indicador de véspera de feriado	0 (não) ou 1 (sim)
<i>Holiday's Bridge</i>	Indicador de "ponte"	0 (não) ou 1 (sim)
<i>Class</i>	Método de previsão com maior precisão	Microsimulação ou ANN

Tabela 5.2: Atributos caracterizadores de cada instância utilizada nos algoritmos k-NN e Árvore de Decisão.

## 5.7 Sumário

Neste capítulo foram apresentados os pormenores relativos à implementação dos três métodos *experts*. Desenvolvidos em linguagem Java, salienta-se a implementação integral do *Q-Learning*, tendo como suporte uma base de dados SQL Server, enquanto que nos restantes módulos foram utilizadas as implementações de k-NN e Árvore de Decisão disponibilizadas na *framework open-source* Weka, através da sua API Java. Para além da descrição das fases de aprendizagem e classificação, é também apresentado o tratamento e transformação de dados efetuado.

## Capítulo 6

# Teste e Avaliação

Este capítulo apresenta os resultados obtidos nesta dissertação e respetiva avaliação, sobretudo os resultados de precisão obtidos pelos métodos de *machine learning* utilizados.

Foram executados dois testes principais. No primeiro, o objetivo foi recolher os dados de precisão e sensibilidade na seleção de métodos de previsão efetuada por cada um dos três *experts*. No segundo, pretende-se sobretudo verificar qual a precisão de previsão das condições de tráfego através dos algoritmos selecionados pelos *experts*, simulando um contexto real.

### 6.1 Dados utilizados para testes

Os testes realizados tiveram como base a utilização dos dados históricos relativos aos primeiros 24 dias de Janeiro de 2014 para 27 *edges* na zona da VCI (Via de cintura interna), no Porto, disponibilizados pela Armis ITS. Estes compreendem dados históricos reais das condições de tráfego divididos por intervalos de 5 minutos, previsões efetuadas, dados característicos de cada dia e dados meteorológicos, entre outros de suporte aos anteriores. São também utilizados os resultados das previsões efetuadas pelos dois algoritmos implementados no sistema NEXT (Microsimulação e ANN) para o mesmo período.

#### 6.1.1 Notas importantes

Importa ainda salientar dois aspetos importantes relativos aos testes efetuados. Primeiro, devido ao elevado tempo de computação necessário para processar as quantidades de dados em questão, para testes foram apenas consideradas as previsões efetuadas a 30min, 1h, 1h30 e 2h, em vez dos 24 períodos disponíveis (até 2 horas em intervalos de 5 minutos). Em segundo, não foi possível obter até ao momento de conclusão desta dissertação dados de incidentes para o período temporal dos restantes dados. A utilização destes dados seria importante para perceber como se comportam as classificações em casos de incidente rodoviário.

## 6.2 Teste 1 - Resultados de seleção do melhor método de previsão de tráfego

Este primeiro teste pretende apresentar os resultados de precisão obtidos nos testes efetuados aos três *experts*. Assim, após treino dos modelos computacionais, estes foram utilizados para selecionar o método de previsão de tráfego ideal para novos estados da rede, testando posteriormente se esse método era o que apresentava um resultado mais próximo da realidade.

No momento de início desta dissertação, a Armis ITS havia já experimentado uma Rede Neuronal Artificial para determinar a escolha do melhor algoritmo de previsão para cada estado da rede e contexto. Assim, mesmo que esta tentativa tenha sido muito primitiva e pouco refinada, foram também calculadas as precisões com base nesses resultados, servindo como base de comparação para os novos *experts*.

### 6.2.1 Resultados utilizando Rede Neuronal Artificial

A tabela 6.1, apresentada de seguida, revela a matriz de confusão resultante do cálculo da quantidade de seleções de métodos de previsão corretas e incorretas realizadas pela Rede Neuronal Artificial inicialmente testada na Armis.

	Microsimulação	ANN	<i>Class Precision</i>
Pred. Microsimulação	104	0	100%
Pred. ANN	63178	15614	19.8%
<i>Class Recall</i>	0.16%	100%	

Tabela 6.1: Matriz de confusão dos resultados de seleção de método de previsão pela técnica Rede Neuronal Artificial.

Observando então a matriz de confusão, verificamos que todas as escolhas do método Microsimulação foram acertadas mas que em contrapartida corresponde apenas a 0,16% das instâncias que deveriam ser classificadas como Microsimulação. Relativamente às previsões de ANN, apenas 19,8% foram corretas. A precisão calculada a partir destes valores, que representa a proporção de todas as classificações que estão corretas, é de **19,9%**. Estes resultados sugerem que o problema a ser explorado nesta dissertação é real e os métodos desenvolvidos terão o intuito de obter melhores resultados.

### 6.2.2 Resultados do expert *Q-Learning*

Após a aprendizagem efetuada na fase de *exploration* do método *Q-Learning*, foram realizados testes de seleção de métodos de previsão para dois dias, um domingo e uma quinta-feira, e avaliada a sua precisão na escolha do algoritmo com melhor previsão nos casos testados. Os dados apresentados de seguida na tabela 6.2, que representa a matriz de confusão, demonstram esses resultados.

	Microsimulação	ANN	<i>Class Precision</i>
Pred. Microsimulação	26 324	794	97,1%
Pred. ANN	331	2503	88,3%
<i>Class Recall</i>	98,8%	75,9%	

Tabela 6.2: Matriz de confusão dos resultados de seleção de método de previsão pela técnica *Q-Learning*.

A precisão obtida a partir destes resultados é de **96,2%**, que representa a percentagem de previsões corretas. Denota-se que as classificações de Microsimulação apresentam resultados percentuais de acerto elevados. No entanto, das classificações de ANN, 88,3% encontram-se corretas e em apenas 75,9% dos casos em que ANN seria o melhor método de previsão, este foi efetivamente escolhido.

### 6.2.3 Resultados do *expert* CBR

Dado que existe um modelo de aprendizagem para cada *edge*, foram realizados testes de precisão e geradas matrizes de confusão para as mesmas. Embora possam ser verificadas no anexo A as matrizes de confusão individuais, os resultados apresentados de seguida refletem a união dos resultados de todas essas *edges*. Em termos de parametrização, após teste com vários valores em diferentes *edges*, os resultados finais utilizam o valor  $k = 7$ , ou seja, na recuperação de casos semelhantes a classificação é atribuída pela maioria dos 7 casos mais próximos.

A validação foi feita com recurso à técnica *cross-validation*, com 10 validações e com uma seleção estratificada dos subconjuntos.

	Microsimulação	ANN	<i>Class Precision</i>
Pred. Microsimulação	590 475	34 553	94,5%
Pred. ANN	21 178	66 400	75,8%
<i>Class Recall</i>	96,5%	65,8%	

Tabela 6.3: Matriz de confusão dos resultados de seleção de método de previsão pela técnica CBR, utilizando *cross-validation*.

Neste teste efetuado ao *expert* CBR, representado na matriz de confusão da tabela 6.3 a precisão obtida é de **92,2%**. Embora relativamente à Microsimulação as percentagens sejam bastante elevadas nota-se, porém, que apenas em 65,8% dos casos em que melhor algoritmo seria o ANN, estes foram classificados como tal.

### 6.2.4 Resultados do *expert* Árvore de Decisão

Tal como para CBR, a aprendizagem das *edges* também foi individualizada, sendo cada uma representada por uma árvore de decisão. Nesse sentido, os resultados foram igualmente agrupados, embora no anexo B possam ser verificadas as matrizes de confusão individuais, para cada *edge*.

A validação foi feita com recurso à técnica *cross-validation*, com 10 validações e com uma seleção estratificada dos subconjuntos.

O critério de divisão utilizado foi o *gain ratio*, com uma profundidade máxima de 15 níveis, e realizando a poda (ou *pruning*) da árvore com um grau de confiança de 0.25. Estes parâmetros foram fixados após estudo do comportamento e precisão com a variação dos mesmos para algumas *edges*.

	Microsimulação	ANN	<i>Class Precision</i>
Pred. Microsimulação	589 713	36 667	94,1%
Pred. ANN	21 940	64 286	74,6%
<i>Class Recall</i>	96,4%	63,7%	

Tabela 6.4: Matriz de confusão dos resultados de seleção de método de previsão pela técnica Árvore de Decisão, utilizando *cross-validation*.

No teste, representado na tabela 6.4, a precisão, ou seja, a percentagem de classificações corretas, é de **91,8%**. Enquanto que a previsão de microsimulação demonstra uma precisão alta, a precisão e a proporção de casos ANN classificados corretamente é inferior.

### 6.2.5 Discussão de Resultados

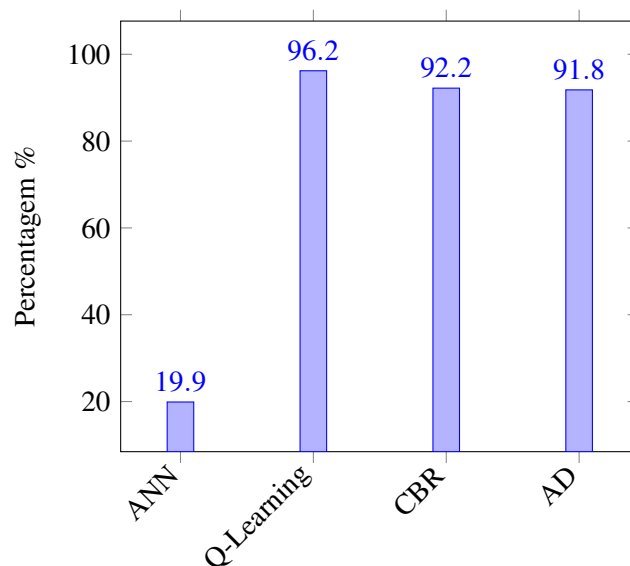


Figura 6.1: Gráfico de resultados de precisão para o teste 1.

Após a análise dos resultados relativos ao desempenho dos *experts* neste primeiro teste, e que podem ser observados graficamente na figura 6.1, é possível retirar algumas conclusões. Observando as matrizes de confusão, percebe-se que os resultados não diferem muito entre os vários métodos utilizados e, embora o método *Q-Learning* pareça destacar-se um pouco, é notória a proporcionalidade entre os resultados de precisão na seleção de Microsimulação e ANN. Nesse

sentido, observam-se dois desempenhos distintos: a precisão de seleção do método Microsimulação e a sensibilidade na sua escolha (número de casos em que o melhor método é Microsimulação e é classificado como tal pelo *expert*) são elevados. No entanto, é claro um desempenho inferior na seleção do método ANN, comparativamente ao anterior.

### 6.3 Teste 2 - Resultados globais de previsão utilizando agentes *experts*

Como abordado anteriormente, embora o intuito desta dissertação seja melhorar a seleção do algoritmo apropriado para cada estado da rede, o objetivo final do sistema onde este módulo se engloba é fornecer previsões acertadas. Este segundo teste teve assim a intenção de perceber como se comportam os resultados de previsão pelo método selecionado. Para tal, dos 24 dias de dados disponíveis, foram selecionados dois dias a serem utilizados para teste, 19 de Janeiro (domingo) e 23 de janeiro (quinta-feira). Os restantes dias foram utilizados para treino dos modelos. De salientar que foram utilizados os mesmos dados de teste em todas as experiências desta secção, para permitir comparações justas. Os resultados são apresentados de seguida para os três modelos, *Q-Learning*, CBR e Árvores de decisão, permitindo a comparação entre os resultados de seleção dos *experts* e a precisão da previsão efetuada pelos métodos por si selecionados. Esta precisão é definida com base no acerto no *traffic status* ou na velocidade média com uma margem de erro de 10%.

#### 6.3.1 Precisão máxima possível

Para ser possível perceber a dimensão da percentagem de acerto, calculou-se também a percentagem de acerto que existiria caso fosse sempre selecionado o melhor método de previsão. Assim, sabemos que essa percentagem representa o valor máximo que pode ser atingido pelos *experts*, permitindo efetuar uma melhor análise global.

	Corretas	Incorretas
Previsões de Tráfego	27139	2813
Percentagem	90,6%	9,4%

Tabela 6.5: Tabela com resultado das melhores previsões efetuadas pelos métodos disponíveis no sistema NEXT.

Observando os valores da tabela 6.5, entende-se que, para o conjunto de casos testados, caso o *expert* acertasse na escolha dos métodos de previsão em todos os casos, a precisão total garantida pela atuação dos métodos de previsão escolhidos seria de **90,6%**. Para os restantes 9,4%, nenhum dos dois métodos de previsão presentes apresentaria uma previsão correta.

#### 6.3.2 *Q-Learning*

Observando a matriz de confusão da tabela 6.6 e as percentagens de previsões corretas apresentados na tabela 6.7, verifica-se que a precisão total do *expert* em termos de seleção do melhor

## Teste e Avaliação

	Microsimulação	ANN	<i>Class Precision</i>
Pred. Microsimulação	26 324	794	97,1%
Pred. ANN	331	2503	88,3%
<i>Class Recall</i>	98,8%	75,9%	<b>Precisão Total = 96,2%</b>

Tabela 6.6: Matriz de confusão dos resultados de seleção de método de previsão pelo algoritmo *Q-Learning*, testado para dias 19 e 23 de Janeiro de 2014.

	Corretas	Incorretas
Previsões de Tráfego	24665	5287
Percentagem	82,3%	17,7%

Tabela 6.7: Tabela com resultado das previsões efetuadas pelo método de previsão selecionado pelo algoritmo *Q-Learning*, testado para dias 19 e 23 de Janeiro de 2014.

método situa-se nos 96,2%, enquanto que, como resultado final do sistema, as previsões corretas correspondem a 82,3% do total de casos testados.

### 6.3.3 CBR

	Microsimulação	ANN	<i>Class Precision</i>
Pred. Microsimulação	25607	1013	96,2%
Pred. ANN	813	2519	75,6%
<i>Class Recall</i>	96,9%	71,3%	<b>Precisão Total = 93,9%</b>

Tabela 6.8: Matriz de confusão dos resultados de seleção de método de previsão pela técnica CBR, testado para dias 19 e 23 de Janeiro de 2014.

	Corretas	Incorretas
Previsões de Tráfego	24945	5007
Percentagem	83,3%	17,7%

Tabela 6.9: Tabela com resultado das previsões efetuadas pelo método de previsão selecionado pela técnica CBR, testado para dias 19 e 23 de Janeiro de 2014.

Para o *expert* CBR, a precisão total de seleção do melhor método situa-se nos 93,9%, enquanto que as previsões de tráfego corretas correspondem a 83,3%.

### 6.3.4 Árvore de Decisão

Para o módulo *expert* baseado em Árvores de Decisão, a precisão total de seleção do melhor método situa-se nos 94,4%, enquanto que as previsões de tráfego corretas correspondem a 83,2%.



## Teste e Avaliação

	Microsimulação	ANN	<i>Class Precision</i>
Pred. Microsimulação	25817	1072	96%
Pred. ANN	603	2460	80%
<i>Class Recall</i>	97,7%	69,6%	<b>Precisão Total = 94,4%</b>

Tabela 6.10: Matriz de confusão dos resultados de seleção de método de previsão pela técnica Árvore de Decisão, testado para dias 19 e 23 de Janeiro de 2014.

	Corretas	Incorretas
Previsões de Tráfego	24918	5034
Porcentagem	83,2%	16,8%

Tabela 6.11: Tabela com resultado das previsões efetuadas pelo método de previsão selecionado pela técnica Árvore de Decisão, testado para dias 19 e 23 de Janeiro de 2014.

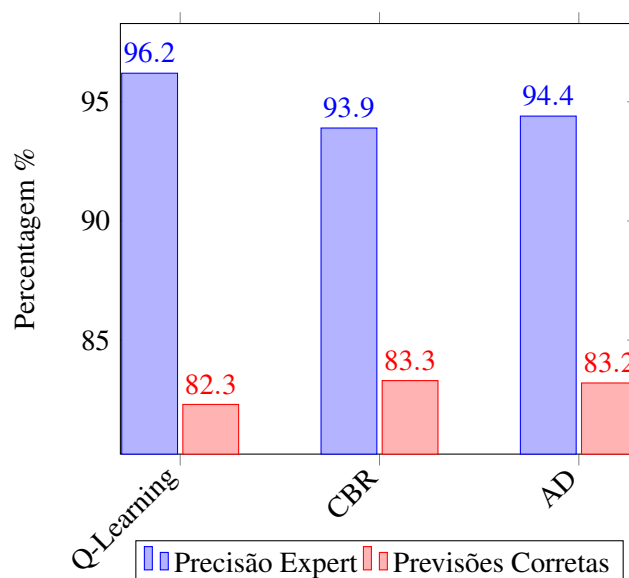


Figura 6.2: Gráfico de resultados de precisão para o teste 2.

### 6.3.5 Discussão de Resultados

Os resultados obtidos neste segundo teste, representados graficamente na figura 6.2, que tem como intuito perceber qual o desempenho total do sistema de previsão de tráfego com a utilização dos métodos *experts*, apresentam-se bastante semelhantes. A percentagem de acerto dos três situa-se nos 82-83%, não distinguindo claramente nenhum dos modelos. Este teste permite, essencialmente, comparar a precisão dos *experts* na seleção do melhor método com a precisão dos métodos nas suas previsões de tráfego. Observa-se assim que os *experts* apresentam percentagens de acerto no melhor método entre os 93% e os 96%, enquanto que a previsão efetuada pelos métodos escolhidos se situa, como referido, entre os 82% e os 83%. Contudo, através dos resultados obtidos na subsecção 6.3.1, entende-se que, como o máximo que se poderia atingir seriam 90,6%, os valores conseguidos situam-se relativamente próximos deste máximo.

## 6.4 Análise dos resultados obtidos

Depois dos testes efetuados e descritos no decorrer deste capítulo, para além da análise já realizada aos resultados de cada um dos dois testes, importa salientar algumas conclusões:

- Em termos de resultados de precisão na seleção do melhor método de previsão, o *expert Q-Learning* destaca-se ligeiramente com 96% de acerto, contra cerca de 92% dos restantes.
- Relativamente às previsões de condições de tráfego realizadas com suporte dos *experts*, todas as soluções apresentam uma precisão idêntica.
- Apesar dessa precisão se situar apenas entre os 82 e 83%, após análise verificou-se que o máximo atingível seria 90,6%, ou seja, mesmo que os *experts* acertassem em todas as seleções do método ideal, esta seria a percentagem máxima atingida para a precisão das previsões de tráfego resultantes.
- A precisão de classificação para Microsimulação é mais elevada do que para ANN. No entanto, como analisado anteriormente, mais de 86% dos casos são previstos com maior exatidão pela Microsimulação, o que reduz bastante o número de casos de ANN utilizados em aprendizagem. Aliado à escassez de dados já atravessada, pode tratar-se de má calibração da aprendizagem para esse método.
- No seguimento do ponto anterior, conclui-se que a maior dificuldade percecionada atualmente pelos modelos se prende com a identificação dos casos em que o método ANN apresenta maior precisão.

## Capítulo 7

# Conclusões e Trabalho Futuro

Neste capítulo, pretende-se apresentar as principais conclusões obtidas referentes ao trabalho desenvolvido durante esta dissertação. Inicialmente, são apresentadas algumas considerações e um breve resumo dos resultados obtidos, assim como uma análise dos mesmos e a discussão sobre a satisfação dos objetivos propostos. Posteriormente, discutida a contribuição deste projeto para a área de previsão de condições de tráfego e especialmente para o sistema NEXT da Armis, no âmbito da qual este projeto foi desenvolvido. Por fim, são apresentadas algumas propostas de trabalho que poderia ser realizado no futuro com vista à melhoria dos módulos desenvolvidos.

### 7.1 Satisfação dos objetivos

Durante a realização desta dissertação foi estudado e desenvolvido um conjunto de módulos inteligentes, apelidados de *experts*, que permitem selecionar o método de previsão de tráfego mais adequado para cada estado da rede, caracterizado pelo local e contexto atual. Desenvolvido no âmbito do sistema NEXT pertencente à Armis ITS, onde neste momento existem dois métodos de previsão funcionais, Microsimulação e Rede Neuronal Artificial (ANN), este projeto teve como intuito preencher uma parte fulcral na informação das condições de trânsito ao utilizador final. Embora o NEXT já permita realizar previsões, independentemente do seu grau de precisão, é necessário saber qual o método indicado para cada situação, assumindo que para situações diferentes o algoritmo com melhor previsão também será diferente. Na análise descritiva dos dados provou-se que embora a Microsimulação apresente previsões mais precisas na maioria das situações (86,4%), existe ainda uma porção importante de casos em que a Rede Neuronal Artificial possui melhor desempenho.

Durante este processo foram então abordados quatro métodos de aprendizagem diferentes: *Q-Learning*, SARSA, *Case-based Reasoning* (CBR) e Árvores de Decisão. Durante a conceptualização dos mesmos, o SARSA acabou por ser descartado pois aplicando-o ao domínio do problema, dado que apenas é considerada uma transição entre estados, ficaria com um comportamento exatamente igual ao *Q-Learning*. Assim, destes dois algoritmos de aprendizagem por reforço, foi desenvolvido o *Q-Learning* de forma integral.

Relativamente à metodologia CBR, esta foi implementada com recurso ao algoritmo k-NN, que permite indexar casos novos e recuperar casos passados, constituídos neste caso pelas características do estado da rede e classificador do melhor método de previsão para esse estado. Assim, na presença de um novo estado da rede, são procurados na base de conhecimento casos semelhantes e respetivo método com melhor performance, que permitirão decidir qual o método de previsão a aplicar ao novo caso.

Por último, foi desenvolvido um módulo constituído por Árvores de Decisão. Utilizando a base de dados fornecida pela Armis, foi possível criar árvores individuais para cada *edge*, que servirão, tal como os modelos anteriores, para seleccionar em tempo real, dependendo das características do estado da rede, o método de previsão ideal.

Em suma, encontram-se desenvolvidos e testados três opções para executar a tarefa de, em tempo real, e perante os novos estados da rede, escolher o algoritmo de previsão adequado e utilizá-lo para garantir informação ao utilizador final.

Em termos de performance, o método *Q-Learning* destacou-se ligeiramente dos restantes com mais de 96% de precisão. No entanto, no que diz respeito às previsões efetuadas a partir dos métodos seleccionados pelos *experts*, todos apresentaram uma precisão idêntica. Tal facto parece indicar que o aperfeiçoamento das técnicas de previsão, em parte com a utilização de mais dados, poderá contribuir para a melhoria dos resultados de todo o sistema. Porém, denota-se também uma maior precisão na seleção do método Microsimulação, mas que é justificável pois em mais de 86% dos casos históricos disponibilizados esse método de previsão apresenta melhor desempenho relativamente ao método ANN.

Considera-se, após estas considerações, que os objetivismos gerais propostos foram atingidos. Após um teste de eficácia com recurso a mais dados e consequente aprovação, uma destas três opções poderá ser escolhida e integrada no sistema NEXT como o novo módulo *expert*.

## 7.2 Trabalho futuro

Analisando o trabalho desenvolvido e resultados obtidos, retiram-se algumas conclusões sobre o que poderá ser realizado no futuro com vista a melhorar a qualidade e utilidade dos modelos desenvolvidos.

Em primeiro lugar, seria importante obter mais dados que permitissem tanto melhorar a aprendizagem dos modelos como testar e obter resultados com maior confiança. Para a aprendizagem foram utilizados os dados históricos de previsões fornecidos, mas que apenas incluem 24 dias. O conjunto de dados ideal para esta dissertação seria o histórico de um ano completo, sendo utilizado outro ano completo para teste e avaliação dos modelos. Para além da escassez destes dados, existem outros dados pouco precisos ou inexistentes. Relativamente aos incidentes, que são dos eventos com maiores consequências na qualidade do fluxo de tráfego, não existiram dados para serem utilizados atempadamente. Quanto às condições meteorológicas, nos dados trabalhados apenas existem dois estados, tempo seco ou chuva mas, embora sejam os mais importantes, há

um conjunto de outras informações que poderiam ser relevantes, como por exemplo, a intensidade da chuva. Neste aspeto, os dados meteorológicos mais recentes obtidos pelo sistema NEXT já incluem um conjunto mais alargado e específico. Por fim, uma fase importante e que poderá constituir um passo enorme na qualidade das previsões, é a utilização de informação relativa ao relacionamento entre *edges* contíguas. Como conclusão, os *experts* desenvolvidos poderão evoluir a par da evolução dos métodos de previsão.

Quanto aos módulos *expert* implementados, existem sempre melhoramentos possíveis. Em primeiro lugar, com a ajuda de um especialista na área de previsão, seria fundamental seleccionar as características do estado da rede consoante a sua influência na possível alteração do *traffic status*, e consequente influência na previsão e escolha dos algoritmos apropriados. Supondo que, por exemplo, a ocorrência de incidentes tem mais peso do que as condições meteorológicas, essas diferenças poderiam ser incluídas nos algoritmos de seleção. Em segundo lugar, poderia ser vantajosa a utilização de técnicas de *meta-learning* [LBG15, VD02] para melhorar a eficácia dos *experts* baseados em Árvores de Decisão e k-NN (CBR). Em ambos os métodos, visto que se utilizou uma instância do mesmo modelo para cada *edge*, as parametrizações foram também iguais para todas elas. Avaliando os resultados individuais, percebemos que existem diferenças de precisão, em alguns casos consideráveis. Com a utilização de técnicas de *meta-learning* poderia ser possível realizar uma aprendizagem da parametrização adequada em função dos dados e comportamento de cada *edge*. Por último, nesta dissertação, a comparação entre os estados da rede previstos e a realidade foi realizada com base no *traffic status* e na velocidade média, do qual a primeira depende neste momento. Seria importante a realização de um estudo aprofundado sobre a melhor forma de calculo do *traffic status* tendo em conta as três métricas de tráfego fundamentais: velocidade, ocupação e volume.

## Conclusões e Trabalho Futuro

# Referências

- [ACMR06] Warren Armstrong, Peter Christen, Eric McCreath e Alistair Rendell. Dynamic Algorithm Selection Using Reinforcement Learning. *2006 International Workshop on Integrating AI and Data Mining*, pages 18–25, 2006. doi:10.1109/AIDM.2006.4.
- [AFR08] Paulo Alexandre, Fonseca Ferreira e Rosaldo Rossetti. Specification and Implementation of an Artificial Transport System. 2008.
- [Alt92] N. S. Altman. An introduction to kernel and nearest-neighbor non-parametric regression. *The American Statistician*, 46(3):175–185, 1992. doi:10.1080/00031305.1992.10475879.
- [And12] Ole Johan Andersen. *How Case-based Reasoning can be used to predict and improve Traffic Flow in Urban Intersections*. PhD thesis, 2012. URL: <http://hdl.handle.net/11250/253148>.
- [AP94] Agnar Aamodt e Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994. doi:10.1.1.56.4481.
- [Arm16] Armis ITS. NEXT, 2016. URL: <http://its.armis.pt/areas/next/>.
- [Bar15a] Joaquim Barros. *Caraterização de estradas em tempo real através de métodos preditivos*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, 2015.
- [BAR15b] Joaquim Barros, Miguel Araujo e Rosaldo J F Rossetti. Short-term real-time traffic prediction methods : a survey. *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, (3-5. June):3–5, 2015.
- [BFAC13] Amna Bouhana, Afef Fekih, Mourad Abed e Habib Chabchoub. An integrated case-based reasoning approach for personalized itinerary search in multimodal transportation systems. *Transportation Research Part C: Emerging Technologies*, 31:30–50, 2013. URL: <http://dx.doi.org/10.1016/j.trc.2013.02.014>, doi:10.1016/j.trc.2013.02.014.
- [BLZH15] Nie Bin, Pi Li, Fan Zhengwei e Yang Huiyuan. Study on Transportation Planning System: Based on Case-Based Reasoning. *The Open Cybernetics & Systemics Journal*, 9:1399–1402, 2015.
- [BSM00] T Bellemans, B De Schutter e B De Moor. On Data Acquisition, Modeling and Simulation of Highway Traffic. *Proc. of the 9th IFAC Symposium in Transportation Systems*, 19:22–27, 2000.

## REFERÊNCIAS

- [ECT13] Directorate General for Mobility European Commission e Transport. Datex II User Guide, 2013. URL: <http://www.datex2.eu/content/datex-background>.
- [FERO08] Paulo A F Ferreira, Edgar F. Esteves, Rosaldo J F Rossetti e Eugênio C. Oliveira. A cooperative simulation framework for traffic and transportation engineering. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008. doi:10.1007/978-3-540-88011-0\_12.
- [FG09] João Filipe e Barreiras Gonçalves. Arquitetura Baseada em Serviços para Redes Veículo-a-Veículo. 2009. doi:oai:digitool.fe.up.pt:14533.
- [Fin98] Eugene Fink. How to Solve It Automatically: Selection Among Problem Solving Methods. *Artificial Intelligence Planning Systems*, pages 128–136, 1998.
- [HFH<sup>+</sup>09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann e Ian H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 2009.
- [HLS07] C.P. Van Hinsbergen, J. Van Lint e F.M. Sanders. Short term traffic prediction models. *14th World Congress on Intelligent Transport Systems: ITS for a better life*, (March):1–18, 2007. URL: <http://trid.trb.org/view.aspx?id=916156>.
- [JM15] Junchen Jin e Xiaoliang Ma. Adaptive Group-Based Signal Control Using Reinforcement Learning with Eligibility Traces. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2015-Octob:2412–2417, 2015. doi:10.1109/ITSC.2015.389.
- [KGM12] Lars Kotthoff, Ian P. Gent e Ian Miguel. An Evaluation of Machine Learning in Algorithm Selection for Search Problems. *AI Communications*, 25(3):257–270, 2012. doi:10.3233/AIC-2012-0533.
- [KLM96] Leslie Pack Kaelbling, Michael L. Littman e Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4(2):237–285, 1996. doi:10.1613/jair.301.
- [KMG06] Lawrence A. Klein, Milton K. Mills e David R.P. Gibson. Traffic detector handbook: Third Edition—Volume I. I(October), 2006. URL: <http://www.fhwa.dot.gov/publications/research/operations/its/06108/02.cfm>.
- [Koh95] Ron Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence*, 14(12):1137–1143, 1995. doi:10.1067/mod.2000.109031.
- [Kol92] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1):3–34, 1992. doi:10.1007/BF00155578.
- [KP98] Ron Kohavi e Foster Provost. Glossary of Terms. *Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, 30(2/3):271–274, 1998. doi:http://dx.doi.org/10.1016/B978-0-08-096682-3.10017-4.
- [KSPRG] Zafeiris Kokkinogenis, Lúcio Sanchez Passos, Rosaldo Rossetti e Joaquim Gabriel. Towards the next-generation traffic simulation tools: a first evaluation.



## REFERÊNCIAS

- [Law01] Klein Lawrence. *Sensor Technologies and Data Requirements for ITS Applications*. 2001.
- [LBG15] Christiane Lemke, Marcin Budka e Bogdan Gabrys. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, 44(1):117–130, 6 2015. doi:10.1007/s10462-013-9406-y.
- [LL00] Michail G. Lagoudakis e Michael L. Littman. Algorithm Selection using Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 511–518, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [LRB09] P. Loureiro, R. Rossetti e R. Braga. Video processing techniques for traffic information acquisition using uncontrolled video streams. *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 127–133, 2009.
- [McC15] James L. McClelland. *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*. 2015.
- [MR06] Tom V. Mathew e K V Krishna Rao. Fundamental relations of traffic flow. *Introduction to Transportation Engineering*, 1:1–8, 2006.
- [Pin14] Tiago Pinho. *Real-time forecasting of traffic conditions on road networks*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, 2014.
- [Pow07] David M W Powers. Evaluation : From Precision , Recall and F-Factor to ROC , Informedness , Markedness & Correlation. (December), 2007.
- [Qui86] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986. doi:10.1023/A:1022643204877.
- [RFBO08] Rosaldo J F Rossetti, Paulo A F Ferreira, Rodrigo A M Braga e Eugênio C. Oliveira. Towards an artificial traffic control system. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2008. doi:10.1109/ITSC.2008.4732719.
- [Ric76] John R. Rice. The Algorithm Selection Problem. *Advances in Computers*, 15(C):65–118, 1976. doi:10.1016/S0065-2458(08)60520-3.
- [RLT11] Rosaldo J F Rossetti, Ronghui Liu e Shuming Tang. Guest editorial special issue on artificial transportation systems and simulation. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):309–312, 2011. doi:10.1109/TITS.2011.2143770.
- [RN94] G Rummary e M Niranjana. On-line Q-learning using connectionist systems. (September):Technical Report CUED/F/INFENG/TR 166, 1994. doi:10.1.1.17.2539.
- [ROB07] Rosaldo J F Rossetti, Eugênio C Oliveira e Ana L C Bazzan. Towards a specification of a framework for sustainable transportation analysis. *13th Portuguese Conference on Artificial Intelligence, Guimarães, Portugal*, 2007.
- [SB98] R S Sutton e a G Barto. *Reinforcement learning: an introduction*, volume 9. The MIT Press, 1998. URL: <https://mitpress.mit.edu/books/reinforcement-learning><https://webdocs.cs.ualberta.ca/~{}sutton/book/ebook/the-book.html>, doi:10.1109/TNN.1998.712192.

## REFERÊNCIAS

- [Sch82] R C Schank. Dynamic memory: a theory of learning in computers and people. *New York: Cambridge University Press*, 1982.
- [Sis12] Virginia P Sisiopiku. Active Traffic Management as a Tool for Addressing Traffic Congestion. *InTech, Intelligent Transportation Systems*, 2012.
- [SJ11] Jiong Song e Zhao Jin. Q-learning based multi-intersection traffic signal control model. *2011 International Conference on System Science, Engineering Design and Manufacturing Informatization, ICSEM 2011*, 2:280–283, 2011. doi:10.1109/ICSSEM.2011.6081298.
- [SL90] S. Rasoul Safavian e David Landgrebe. A Survey of Decsion Tree Clasifier Methodology. *IEEE Transactions on Systems, Man, And Cybernetics*, 21(3):660–674, 1990.
- [SSAEA15] Adel Mounir Said, Ahmed Soua, Emad Abd-Elrahman e Hossam Afifi. Context-aware multi-modal traffic management in ITS: A Q-learning based algorithm. *IWCMC 2015 - 11th International Wireless Communications and Mobile Computing Conference*, pages 674–679, 2015. doi:10.1109/IWCMC.2015.7289164.
- [Ste97] Stephen V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77–89, 10 1997. doi:10.1016/S0034-4257(97)00083-7.
- [Tra09] Transport Advisory Committee. Report on Study of Road Traffic Congestion in Hong Kong. 2009.
- [Uni15] United States Department of Transportation - Federal Highway Administration. Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation. Technical report, 2015. URL: [http://www.ops.fhwa.dot.gov/congestion/\\_report/](http://www.ops.fhwa.dot.gov/congestion/_report/).
- [VD02] Ricardo Vilalta e Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 2002. doi:10.1023/A:1019956318069.
- [Wat89] C J C Watkins. Learning from Delayed Rewards. (JANUARY 1989), 1989.
- [Wat99] I. Watson. Case-based reasoning is a methodology not a technology. *Knowledge-Based Systems*, 12(5-6):303–308, 1999. doi:10.1016/S0950-7051(99)00020-9.
- [WD92] Christopher J. C. H. Watkins e Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992. doi:10.1007/BF00992698.
- [Zai14] Ana Zaiat. Integrated Multimodal Transportation Dashboard. page 103, 2014.
- [ZCWY14] Yu Zheng, Licia Capra, Ouri Wolfson e Hai Yang. Urban Computing: Concepts, Methodologies, and Applications. *ACM Trans. Intell. Syst. Technol.*, 5(3):38:1–38:55, 2014. doi:10.1145/2629592.
- [ZRC14] Ana Zaiat, Rosaldo J. F. Rossetti e Ricardo J. S. Coelho. Towards an integrated multimodal transportation dashboard. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 145–150. IEEE, 10 2014. doi:10.1109/ITSC.2014.6957681.

## Anexo A

# Matrizes de confusão de *cross-validation* para cada *edge* pelo método CBR

As matrizes de confusão apresentadas de seguida revelam a precisão e sensibilidade geradas por *cross-validation* das classificações de métodos de previsão efetuadas pelo método *Case-based reasoning*, para as *edges* com dados disponíveis.

accuracy: 90.68% +/- 0.31% (mikro: 90.68%)

	true micro	true ann	class precision
pred. micro	22855	1577	93.55%
pred. ann	999	2205	68.82%
class recall	95.81%	58.30%	

Figura A.1: Matriz de confusão para a *edge* com ID 80920.

accuracy: 87.57% +/- 0.54% (mikro: 87.57%)

	true micro	true ann	class precision
pred. micro	16473	1654	90.88%
pred. ann	1062	2661	71.47%
class recall	93.94%	61.67%	

Figura A.2: Matriz de confusão para a *edge* com ID 80942.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método CBR

accuracy: 95.15% +/- 0.52% (mikro: 95.15%)

	true micro	true ann	class precision
pred. micro	25987	989	96.33%
pred. ann	350	310	46.97%
class recall	98.67%	23.86%	

Figura A.3: Matriz de confusão para a *edge* com ID 80945.

accuracy: 97.89% +/- 0.13% (mikro: 97.89%)

	true micro	true ann	class precision
pred. micro	26991	484	98.24%
pred. ann	100	61	37.89%
class recall	99.63%	11.19%	

Figura A.4: Matriz de confusão para a *edge* com ID 80948.

accuracy: 97.31% +/- 0.18% (mikro: 97.31%)

	true micro	true ann	class precision
pred. micro	26670	588	97.84%
pred. ann	154	188	54.97%
class recall	99.43%	24.23%	

Figura A.5: Matriz de confusão para a *edge* com ID 81091.

accuracy: 98.04% +/- 0.04% (mikro: 98.04%)

	true micro	true ann	class precision
pred. micro	27076	525	98.10%
pred. ann	16	19	54.29%
class recall	99.94%	3.49%	

Figura A.6: Matriz de confusão para a *edge* com ID 81096.

accuracy: 81.37% +/- 0.79% (mikro: 81.37%)

	true micro	true ann	class precision
pred. micro	17539	3011	85.35%
pred. ann	2132	4918	69.76%
class recall	89.16%	62.03%	

Figura A.7: Matriz de confusão para a *edge* com ID 81099.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método CBR

accuracy: 98.90% +/- 0.05% (mikro: 98.90%)

	true micro	true ann	class precision
pred. micro	27300	269	99.02%
pred. ann	36	31	46.27%
class recall	99.87%	10.33%	

Figura A.8: Matriz de confusão para a *edge* com ID 81211.

accuracy: 96.26% +/- 0.23% (mikro: 96.26%)

	true micro	true ann	class precision
pred. micro	25811	688	97.40%
pred. ann	346	791	69.57%
class recall	98.68%	53.48%	

Figura A.9: Matriz de confusão para a *edge* com ID 81364.

accuracy: 90.95% +/- 0.49% (mikro: 90.95%)

	true micro	true ann	class precision
pred. micro	17220	1259	93.19%
pred. ann	1242	7915	86.44%
class recall	93.27%	86.28%	

Figura A.10: Matriz de confusão para a *edge* com ID 81527.

accuracy: 98.07% +/- 0.08% (mikro: 98.07%)

	true micro	true ann	class precision
pred. micro	27008	446	98.38%
pred. ann	88	94	51.65%
class recall	99.68%	17.41%	

Figura A.11: Matriz de confusão para a *edge* com ID 81530.

accuracy: 97.77% +/- 0.16% (mikro: 97.77%)

	true micro	true ann	class precision
pred. micro	26941	523	98.10%
pred. ann	94	78	45.35%
class recall	99.65%	12.98%	

Figura A.12: Matriz de confusão para a *edge* com ID 81566.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método CBR

accuracy: 97.01% +/- 0.17% (mikro: 97.01%)

	true micro	true ann	class precision
pred. micro	26568	682	97.50%
pred. ann	145	241	62.44%
class recall	99.46%	26.11%	

Figura A.13: Matriz de confusão para a *edge* com ID 81786.

accuracy: 98.25% +/- 0.20% (mikro: 98.25%)

	true micro	true ann	class precision
pred. micro	27000	382	98.60%
pred. ann	101	153	60.24%
class recall	99.63%	28.60%	

Figura A.14: Matriz de confusão para a *edge* com ID 81795.

accuracy: 86.21% +/- 0.81% (mikro: 86.21%)

	true micro	true ann	class precision
pred. micro	8796	1643	84.26%
pred. ann	2167	15030	87.40%
class recall	80.23%	90.15%	

Figura A.15: Matriz de confusão para a *edge* com ID 81801.

accuracy: 96.27% +/- 0.16% (mikro: 96.27%)

	true micro	true ann	class precision
pred. micro	26383	820	96.99%
pred. ann	211	222	51.27%
class recall	99.21%	21.31%	

Figura A.16: Matriz de confusão para a *edge* com ID 81957.

accuracy: 87.51% +/- 0.49% (mikro: 87.51%)

	true micro	true ann	class precision
pred. micro	19789	2044	90.64%
pred. ann	1409	4394	75.72%
class recall	93.35%	68.25%	

Figura A.17: Matriz de confusão para a *edge* com ID 82114.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método CBR

accuracy: 92.55% +/- 0.36% (mikro: 92.55%)

	true micro	true ann	class precision
pred. micro	24861	1488	94.35%
pred. ann	570	717	55.71%
class recall	97.76%	32.52%	

Figura A.18: Matriz de confusão para a *edge* com ID 82123.

accuracy: 93.04% +/- 0.48% (mikro: 93.04%)

	true micro	true ann	class precision
pred. micro	24689	1289	95.04%
pred. ann	634	1024	61.76%
class recall	97.50%	44.27%	

Figura A.19: Matriz de confusão para a *edge* com ID 82130.

accuracy: 89.12% +/- 0.46% (mikro: 89.12%)

	true micro	true ann	class precision
pred. micro	22547	1804	92.59%
pred. ann	1204	2081	63.35%
class recall	94.93%	53.56%	

Figura A.20: Matriz de confusão para a *edge* com ID 82159.

accuracy: 80.79% +/- 0.69% (mikro: 80.79%)

	true micro	true ann	class precision
pred. micro	19315	3556	84.45%
pred. ann	1752	3013	63.23%
class recall	91.68%	45.87%	

Figura A.21: Matriz de confusão para a *edge* com ID 82287.

accuracy: 86.05% +/- 0.61% (mikro: 86.05%)

	true micro	true ann	class precision
pred. micro	20377	2228	90.14%
pred. ann	1626	3405	67.68%
class recall	92.61%	60.45%	

Figura A.22: Matriz de confusão para a *edge* com ID 82290.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método CBR

accuracy: 96.32% +/- 0.25% (mikro: 96.32%)

	true micro	true ann	class precision
pred. micro	25784	632	97.61%
pred. ann	384	836	68.52%
class recall	98.53%	56.95%	

Figura A.23: Matriz de confusão para a *edge* com ID 82476.

accuracy: 90.04% +/- 0.39% (mikro: 90.04%)

	true micro	true ann	class precision
pred. micro	20860	1614	92.82%
pred. ann	1135	3991	77.86%
class recall	94.84%	71.20%	

Figura A.24: Matriz de confusão para a *edge* com ID 82537.

accuracy: 88.86% +/- 0.58% (mikro: 88.86%)

	true micro	true ann	class precision
pred. micro	17861	1696	91.33%
pred. ann	1383	6696	82.88%
class recall	92.81%	79.79%	

Figura A.25: Matriz de confusão para a *edge* com ID 82578.

accuracy: 83.70% +/- 0.66% (mikro: 83.70%)

	true micro	true ann	class precision
pred. micro	17774	2662	86.97%
pred. ann	1838	5326	74.34%
class recall	90.63%	66.68%	

Figura A.26: Matriz de confusão para a *edge* com ID 83102.



## Anexo B

# Matrizes de confusão de *cross-validation* para cada *edge* pelo método Árvore de Decisão

As matrizes de confusão apresentadas de seguida revelam a precisão e sensibilidade geradas por *cross-validation* das classificações de métodos de previsão efetuadas por Árvores de Decisão, para as *edges* com dados disponíveis.

accuracy: 90.59% +/- 0.48% (mikro: 90.59%)

	true micro	true ann	class precision
pred. micro	22695	1441	94.03%
pred. ann	1159	2341	66.89%
class recall	95.14%	61.90%	

Figura B.1: Matriz de confusão para a *edge* com ID 80920.

accuracy: 84.95% +/- 0.67% (mikro: 84.95%)

	true micro	true ann	class precision
pred. micro	16748	2501	87.01%
pred. ann	787	1814	69.74%
class recall	95.51%	42.04%	

Figura B.2: Matriz de confusão para a *edge* com ID 80942.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método Árvore de Decisão

accuracy: 95.23% +/- 0.34% (mikro: 95.23%)

	true micro	true ann	class precision
pred. micro	26071	1053	96.12%
pred. ann	266	246	48.05%
class recall	98.99%	18.94%	

Figura B.3: Matriz de confusão para a *edge* com ID 80945.

accuracy: 97.93% +/- 0.13% (mikro: 97.93%)

	true micro	true ann	class precision
pred. micro	27016	496	98.20%
pred. ann	75	49	39.52%
class recall	99.72%	8.99%	

Figura B.4: Matriz de confusão para a *edge* com ID 80948.

accuracy: 97.07% +/- 0.18% (mikro: 97.07%)

	true micro	true ann	class precision
pred. micro	26726	711	97.41%
pred. ann	98	65	39.88%
class recall	99.63%	8.38%	

Figura B.5: Matriz de confusão para a *edge* com ID 81091.

accuracy: 97.83% +/- 0.13% (mikro: 97.83%)

	true micro	true ann	class precision
pred. micro	27010	518	98.12%
pred. ann	82	26	24.07%
class recall	99.70%	4.78%	

Figura B.6: Matriz de confusão para a *edge* com ID 81096.

accuracy: 80.55% +/- 0.68% (mikro: 80.55%)

	true micro	true ann	class precision
pred. micro	17150	2846	85.77%
pred. ann	2521	5083	66.85%
class recall	87.18%	64.11%	

Figura B.7: Matriz de confusão para a *edge* com ID 81099.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método Árvore de Decisão

accuracy: 98.76% +/- 0.16% (mikro: 98.76%)

	true micro	true ann	class precision
pred. micro	27244	250	99.09%
pred. ann	92	50	35.21%
class recall	99.66%	16.67%	

Figura B.8: Matriz de confusão para a *edge* com ID 81211.

accuracy: 96.10% +/- 0.35% (mikro: 96.10%)

	true micro	true ann	class precision
pred. micro	25856	777	97.08%
pred. ann	301	702	69.99%
class recall	98.85%	47.46%	

Figura B.9: Matriz de confusão para a *edge* com ID 81364.

accuracy: 90.75% +/- 0.56% (mikro: 90.75%)

	true micro	true ann	class precision
pred. micro	17121	1215	93.37%
pred. ann	1341	7959	85.58%
class recall	92.74%	86.76%	

Figura B.10: Matriz de confusão para a *edge* com ID 81527.

accuracy: 98.10% +/- 0.12% (mikro: 98.10%)

	true micro	true ann	class precision
pred. micro	27024	453	98.35%
pred. ann	72	87	54.72%
class recall	99.73%	16.11%	

Figura B.11: Matriz de confusão para a *edge* com ID 81530.

accuracy: 97.69% +/- 0.18% (mikro: 97.69%)

	true micro	true ann	class precision
pred. micro	26891	495	98.19%
pred. ann	144	106	42.40%
class recall	99.47%	17.64%	

Figura B.12: Matriz de confusão para a *edge* com ID 81566.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método Árvore de Decisão

accuracy: 96.82% +/- 0.27% (mikro: 96.82%)

	true micro	true ann	class precision
pred. micro	26521	686	97.48%
pred. ann	192	237	55.24%
class recall	99.28%	25.68%	

Figura B.13: Matriz de confusão para a *edge* com ID 81786.

accuracy: 98.32% +/- 0.16% (mikro: 98.32%)

	true micro	true ann	class precision
pred. micro	26989	352	98.71%
pred. ann	112	183	62.03%
class recall	99.59%	34.21%	

Figura B.14: Matriz de confusão para a *edge* com ID 81795.

accuracy: 85.48% +/- 0.74% (mikro: 85.48%)

	true micro	true ann	class precision
pred. micro	8834	1883	82.43%
pred. ann	2129	14790	87.42%
class recall	80.58%	88.71%	

Figura B.15: Matriz de confusão para a *edge* com ID 81801.

accuracy: 96.10% +/- 0.18% (mikro: 96.10%)

	true micro	true ann	class precision
pred. micro	26432	916	96.65%
pred. ann	162	126	43.75%
class recall	99.39%	12.09%	

Figura B.16: Matriz de confusão para a *edge* com ID 81957.

accuracy: 87.03% +/- 0.63% (mikro: 87.03%)

	true micro	true ann	class precision
pred. micro	19866	2253	89.81%
pred. ann	1332	4185	75.86%
class recall	93.72%	65.00%	

Figura B.17: Matriz de confusão para a *edge* com ID 82114.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método Árvore de Decisão

**accuracy: 92.43% +/- 0.28% (mikro: 92.43%)**

	true micro	true ann	class precision
pred. micro	24896	1557	94.11%
pred. ann	535	648	54.78%
class recall	97.90%	29.39%	

Figura B.18: Matriz de confusão para a *edge* com ID 82123.

**accuracy: 92.88% +/- 0.32% (mikro: 92.88%)**

	true micro	true ann	class precision
pred. micro	24682	1326	94.90%
pred. ann	641	987	60.63%
class recall	97.47%	42.67%	

Figura B.19: Matriz de confusão para a *edge* com ID 82130.

**accuracy: 89.39% +/- 0.33% (mikro: 89.39%)**

	true micro	true ann	class precision
pred. micro	22886	2068	91.71%
pred. ann	865	1817	67.75%
class recall	96.36%	46.77%	

Figura B.20: Matriz de confusão para a *edge* com ID 82159.

**accuracy: 80.81% +/- 0.54% (mikro: 80.81%)**

	true micro	true ann	class precision
pred. micro	19375	3611	84.29%
pred. ann	1692	2958	63.61%
class recall	91.97%	45.03%	

Figura B.21: Matriz de confusão para a *edge* com ID 82287.

**accuracy: 85.10% +/- 0.33% (mikro: 85.10%)**

	true micro	true ann	class precision
pred. micro	20500	2614	88.69%
pred. ann	1503	3019	66.76%
class recall	93.17%	53.59%	

Figura B.22: Matriz de confusão para a *edge* com ID 82290.

Matrizes de confusão de *cross-validation* para cada *edge* pelo método Árvore de Decisão

accuracy: 95.91% +/- 0.27% (mikro: 95.91%)

	true micro	true ann	class precision
pred. micro	25907	868	96.76%
pred. ann	261	600	69.69%
class recall	99.00%	40.87%	

Figura B.23: Matriz de confusão para a *edge* com ID 82476.

accuracy: 89.04% +/- 0.48% (mikro: 89.04%)

	true micro	true ann	class precision
pred. micro	20698	1728	92.29%
pred. ann	1297	3877	74.93%
class recall	94.10%	69.17%	

Figura B.24: Matriz de confusão para a *edge* com ID 82537.

accuracy: 88.92% +/- 0.73% (mikro: 88.92%)

	true micro	true ann	class precision
pred. micro	17774	1592	91.78%
pred. ann	1470	6800	82.22%
class recall	92.36%	81.03%	

Figura B.25: Matriz de confusão para a *edge* com ID 82578.

accuracy: 80.91% +/- 1.02% (mikro: 80.91%)

	true micro	true ann	class precision
pred. micro	16801	2457	87.24%
pred. ann	2811	5531	66.30%
class recall	85.67%	69.24%	

Figura B.26: Matriz de confusão para a *edge* com ID 83102.